

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, информатики и информационных технологий  
Кафедра информатики, информационных технологий  
и методики обучения информатике

# **МЕТОДИКА ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ ИСПОЛНИТЕЛЕЙ С ОБРАТНОЙ СВЯЗЬЮ НА УРОКАХ ИНФОРМАТИКИ В ШКОЛЕ**

*Выпускная квалификационная работа  
по направлению «44.03.01 – Педагогическое образование»,  
профиль «Информатика»*

Работа допущена к защите  
« \_\_\_\_ » \_\_\_\_\_ 2016 г.  
Зав. кафедрой \_\_\_\_\_

Исполнитель: студентка группы БИ-41  
ИМИиИТ  
Доронина К.Е.  
Руководитель: ст. преподаватель кафедры  
ИИТиМОИ  
Шимов И.В.

Екатеринбург – 2016

## Реферат

**Доронина К.Е.** МЕТОДИКА ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ ИСПОЛНИТЕЛЕЙ С ОБРАТНОЙ СВЯЗЬЮ НА УРОКАХ ИНФОРМАТИКИ В ШКОЛЕ, выпускная квалификационная работа: 72 стр., рис. 34, табл. 3, библи. 25 назв., приложений 3.

**Ключевые слова:** алгоритмизация, программирование, исполнители с обратной связью, реальные исполнители, программируемые устройства, робототехнические устройства, комплекты LEGO MINDSTORMS Education EV3, разработка методических рекомендаций.

**Объект исследования:** процесс обучения основам алгоритмизации и программирования на уроках информатики в школе.

**Цель работы:** разработка методики обучения программированию с использованием исполнителей с обратной связью на уроках информатики в школе.

В работе рассматриваются требования ФГОС к школьному курсу информатики, задачи для реализации этих требований, содержание алгоритмической линии в обязательном минимуме по информатике. Описаны результаты анализа учебно-методических комплексов по дисциплине «Информатика и ИКТ», соответствующих ФГОС. Представлен результат сравнительного анализа существующих виртуальных и реальных исполнителей с обратной связью.

В ходе работы были разработаны методические рекомендации по использованию исполнителей с обратной связью при изучении раздела «Алгоритмы и исполнители» и содержание практических занятий.

Разработанная методика прошла апробацию в Уральском государственном педагогическом университете, однако, после некоторой адаптации может быть использована в работе преподавателей информатики любого учебного заведения.

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>ГЛАВА 1.    СОВРЕМЕННЫЕ ПОДХОДЫ К ОБУЧЕНИЮ ОСНОВАМ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ.....</b>	<b>6</b>
1.1. МЕСТО СОДЕРЖАТЕЛЬНОЙ ЛИНИИ «АЛГОРИТМЫ И ИСПОЛНИТЕЛИ» В ФГОС.....	6
1.2. АНАЛИЗ СУЩЕСТВУЮЩИХ УЧЕБНО-МЕТОДИЧЕСКИХ КОМПЛЕКСОВ ПО ДИСЦИПЛИНЕ «ИНФОРМАТИКА И ИКТ» .....	10
1.3. ОБЗОР СУЩЕСТВУЮЩИХ ИСПОЛНИТЕЛЕЙ С ОБРАТНОЙ СВЯЗЬЮ И МЕТОДИКА ИХ ИСПОЛЬЗОВАНИЯ.....	16
<b>ГЛАВА 2.    РАЗРАБОТКА МЕТОДИКИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ ИСПОЛНИТЕЛЕЙ С ОБРАТНОЙ СВЯЗЬЮ .....</b>	<b>32</b>
2.1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ ИСПОЛНИТЕЛЕЙ С ОБРАТНОЙ СВЯЗЬЮ ПРИ ИЗУЧЕНИИ РАЗДЕЛА «АЛГОРИТМЫ И ИСПОЛНИТЕЛИ».....	32
2.2. СОДЕРЖАНИЕ УРОКОВ-ПРАКТИК С ИСПОЛЬЗОВАНИЕМ ИСПОЛНИТЕЛЕЙ С ОБРАТНОЙ СВЯЗЬЮ .....	50
2.3. АПРОБАЦИЯ РАЗРАБОТАННЫХ МАТЕРИАЛОВ .....	61
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>65</b>
<b>БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....</b>	<b>67</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>70</b>

## Введение

Жизнь в современном обществе выдвигает новые требования к социализации учащихся и их профессиональной ориентации. В повседневном окружении человека стало появляться все больше разнообразных программируемых устройств, например, бытовая техника, кофейные автоматы, телевизоры, терминалы для оплаты услуг и многое другое. Широкий круг использования информационных систем и технологий приводит к тому, что множество профессий компьютеризируется.

Для успешного использования компьютерных технологий, а также для решения жизненных задач, человеку необходимо проанализировать полученную задачу; формализовать её условие; составить план действий и уметь скорректировать его в случае обнаружения ошибки. Без владения логическими методами познания, такими как анализ, синтез, сравнительная оценка, обобщение и абстрагирование, а также несформированным алгоритмическим мышлением выпускник школы окажется не готовым к современным условиям жизни [2].

Ориентируясь в первую очередь на потребности семьи, личность ребенка, а также на запросы общества и государства, был разработан федеральный государственный образовательный стандарт нового поколения. Введение ФГОС по предмету «Информатика и ИКТ» привело к пересмотру содержания обучения. Первостепенным осталось освоение информационно-коммуникационных технологий. Но также внимание уделяется и на формирование компетентности в области алгоритмизации и программирования [19]. Последнее обусловило **актуальность** разработки методик, с использованием новых средств обучения.

**Объект исследования:** процесс обучения основам алгоритмизации и программирования на уроках информатики в школе.

**Предмет исследования:** использование исполнителей с обратной связью при обучении программированию на уроках информатики и ИКТ.

**Цель исследования:** разработка методики обучения программированию с использованием исполнителей с обратной связью на уроках информатики в школе.

**Задачи исследования:**

1. Определить место содержательной линии «Алгоритмизация и программирование» в ФГОС.
2. Провести сравнительный анализ существующих учебно-методических комплексов по дисциплине «Информатика и ИКТ».
3. Провести сравнительный анализ виртуальных и реальных исполнителей алгоритмов.
4. Разработать методические рекомендации по использованию реальных исполнителей в рамках содержательной линии «Алгоритмизация и программирование» дисциплины «Информатика и ИКТ».
5. Разработать содержание уроков-практик для обучения программированию с использованием управляемых устройств из набора LEGO MINDSTORMS Education EV3.
6. Провести апробацию разработанных материалов.

# **Глава 1. Современные подходы к обучению основам алгоритмизации и программирования**

## ***1.1. Место содержательной линии «Алгоритмы и исполнители» в ФГОС***

Введение нового Федерального государственного образовательного стандарта общего образования привело к пересмотру содержания обучения во всех дисциплинах, в том числе и в рамках дисциплины «Информатика и ИКТ» в 8-9 классах.

Приоритетными объектами изучения в курсе информатики основной школы остаются информационные процессы и информационные технологии. В теоретической части курса раскрывается содержание информационной технологии решения задачи. Рассматриваются обобщающие понятия – информационный процесс, информационная модель и информационные основы управления.

Практическая часть курса направлена на освоение учащимися навыков использования средств информационных технологий, которые являются значимыми не только для формирования компьютерной грамотности, социализации школьников и последующей деятельности выпускников, но и для повышения эффективности освоения других учебных предметов [19].

Изучение информатики и информационных технологий в основной школе направлено на достижение следующих целей:

- освоение знаний, составляющих основу научных представлений об информации, информационных процессах, системах, технологиях и моделях;
- овладение умениями работать с различными видами информации с помощью компьютера и других средств информационных и коммуникационных технологий, организовывать собственную информационную деятельность и планировать ее результаты;
- развитие познавательных интересов, интеллектуальных и творческих способностей средствами ИКТ;

- воспитание ответственного отношения к информации с учетом правовых и этических аспектов ее распространения; избирательного отношения к полученной информации;
- выработка навыков применения средств ИКТ в повседневной жизни, при выполнении индивидуальных и коллективных проектов, в учебной деятельности, дальнейшем освоении профессий, востребованных на рынке труда [19].

Федеральный базисный учебный план для образовательных учреждений Российской Федерации отводит 105 часов для обязательного изучения информатики и информационно-коммуникационных технологий на ступени основного общего образования. В том числе в VIII классе – 35 учебных часов из расчета одного учебного часа в неделю и IX классе – 70 учебных часов из расчета двух учебных часов в неделю. В примерной программе предусмотрен резерв свободного учебного времени в объеме одиннадцати часов для реализации авторских подходов; использования разнообразных форм организации учебного процесса; внедрения современных методов обучения и педагогических технологий; учета региональных условий [14].

В примерной программе по дисциплине «Информатика и ИКТ» на раздел «Алгоритмы и исполнители» отводится 19 часов – это 20% от общего времени, выделяемого на дисциплину [14].

Соотношение удельных весов различных разделов содержания по дисциплине «Информатика и ИКТ» представлено в Таблице 1.

В ФГОС и обязательном минимуме по информатике содержание алгоритмической линии определяется через следующий перечень понятий: центральное теоретическое понятие – алгоритм, вводится как содержательное понятие, свойства алгоритмов, исполнители алгоритмов, система команд исполнителя; формальное исполнение алгоритмов; основные алгоритмические конструкции; вспомогательные алгоритмы [14].

Таблица 1.

Соотношение удельных весов различных разделов содержания по дисциплине «Информатика и ИКТ»

	1. Информационные процессы						2. Информационные технологии					
	Информация и информационные процессы	Представление информации	Компьютер как универсальное устройство обработки информации	Алгоритмы и исполнители	Формализация и моделирование	Информационные процессы и технологии в обществе	Обработка текстовой информации	Обработка графической информации	Мультимедийные технологии	Обработка числовой информации	Хранение информации	Коммуникационные технологии
Общее число часов – 105 Резерв времени: 11 часов (10, 5%)												
Учебные часы	4	6	4	19	8	4	14	4	8	6	4	12
Проценты	4,2%	6%	4,2%	20%	9%	4,2%	15%	4,2%	9%	6%	4,2%	13%

Понятия управления и обратной связи вводятся в контексте работы с компьютером. Также данные понятия переносятся и в более широкий контекст социальных, технологических и биологических систем. Он поддержан построением программ управления движущимися объектами в виртуальных и реальных средах.

Резерв времени отводится на разработку алгоритмов, решающих поставленные задачи с использованием математических функций для записи арифметических выражения, операторов ветвления и цикла [14].

Данный раздел реализуется в изучении таких образовательных областей, как информатика и информационные технологии, математика и естествознание.

В соответствии с ФГОС изучение информатики в основной школе должно обеспечить [19]:

- формирование информационной и алгоритмической культуры;
- формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;



- формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойствах;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами – линейной, условной и циклической;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей – таблицы, схемы, графики, диаграммы, с использованием соответствующих программных средств обработки данных;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

Исходя из требования ФГОС, необходимо решить одну из важнейших проблем образования – отбор средств обучения. Процесс вхождения школы в современное мировое образовательное пространство требует совершенствования компьютерно-информационной составляющей. Особый интерес представляют вопросы, связанные с компьютеризацией обучения, поскольку традиционные методы без использования технических средств давно исчерпали свои возможности.

Актуальной в данном вопросе будет реализация обучения с использованием образовательной среды исполнителей с обратной связью.

Главная цель внедрения реальных исполнителей в образовательную деятельность школ – это создание благоприятных условий для разностороннего развития личности, которое включает в себя интеллектуальное развитие,

удовлетворение интересов, выявление склонностей, развитие способностей обучающихся, их самообразование и профессиональное самоопределение [3].

## ***1.2. Анализ существующих учебно-методических комплексов по дисциплине «Информатика и ИКТ»***

В настоящее время по дисциплине «Информатика и ИКТ» имеются значительные учебно-методические наработки для разных возрастных групп учащихся [20]. Издано множество учебников, учебных пособий и методических рекомендаций для преподавателей.

Одной из задач основного курса является обучение основам алгоритмизации и программирования, направленное на развитие алгоритмического, логического мышления учеников, а также на формирование операционного типа мышления. Методика обучения основам алгоритмизации и программирования представлена в рамках курсов известных авторских коллективов. Рассмотрим подробнее курсы этих авторов.

### **УМК «Информатика и ИКТ», автор Семакин И.Г. 7-9 классы, основная школа**

Данный учебно-методический комплекс, обеспечивающий обучение курсу информатики, соответствует ФГОС и включает в себя [12]:

- учебник «Информатика» для 7 класса. Авторы: Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В.;
- учебник «Информатика» для 8 класса. Авторы: Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В.;
- учебник «Информатика» для 9 класса. Авторы: Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В.;
- задачник-практикум под редакцией Семакина И.Г., Хеннера Е.К.;
- методическое пособие для учителя. Авторы: Семакин И.Г., Шеина Т.Ю.;
- комплект цифровых образовательных ресурсов, помещенный в Единую коллекцию ЦОР [4];

- комплект дидактических материалов для текущего контроля результатов обучения информатике в основной школе, под редакцией Семакина И.Г.

Данный курс опирается на базовые научные представления предметной области: информация, информационные процессы, информационные модели [13].

Большое внимание в курсе уделено формированию алгоритмической культуры учащихся, развитию алгоритмического мышления, входящим в перечень предметных результатов ФГОС. Этой теме посвящена большая часть содержания и учебного планирования в 9 классе.

Для практической работы используется исполнитель алгоритмов, разработанный авторами и входящий в комплект ЦОР – «Стрелочка». Для изучения основ программирования используется язык Паскаль [11].

На тему «Управление и алгоритмы» в данной авторской программе отводится 12 часов. В курсе предполагается решение большого количества задач, позволяющих усвоить учащимся основы алгоритмизации и программирования на высоком уровне. Задачник-практикум дает обширный материал для организации практической работы на уроках и домашней работы учащихся. Большое число разнообразных заданий предоставляет учителю возможность варьировать содержание курса по времени и уровню сложности.

При изучении данного курса учащиеся смогут: выполнять трассировку заданных простых алгоритмов; строить блок-схемы несложных алгоритмов; использовать школьный алгоритмический язык для описания алгоритмов; работать с готовой программой на одном из языков программирования высокого уровня; составлять несложные программы решения вычислительных задач; осуществлять отладку и тестирование программы.

**УМК «Информатика и ИКТ», автор Угринович Н.Д.  
8-9 классы, основная школа**

Состав УМК:

- учебник «Информатика и ИКТ. Базовый курс», 8 класс;
- учебник «Информатика и ИКТ. Базовый курс», 9 класс;
- практикум по информатике и информационным технологиям, 8-11 классы;
- методическое пособие для учителя «Информатика и ИКТ. Методическое пособие», 8-11 классы;
- комплект плакатов, «Информатика и ИКТ. Основная школа»;
- методические рекомендации по использованию плакатов «Информатика и ИКТ. Основная школа»;
- авторская мастерская Угриновича Н.Д.;
- ЭОР клавиатурный тренажер «Руки солиста».

Особое место в данном комплексе по дисциплине «Информатика и ИКТ» для 9 класса занимает тема «Алгоритмизация и основы объектно-ориентированного программирования» [16]. В этой теме рассматриваются все основные алгоритмические структуры и их кодирование на трех языках программирования:

- алгоритмическом языке OpenOffice Basic, который входит в свободно распространяемое интегрированное офисное приложение OpenOffice Basic в операционных системах Windows и Linux;
- объектно-ориентированном языке Visual Basic 2005, который распространяется по лицензии корпорации Microsoft;
- объектно-ориентированном языке Gambas (аналог – Visual Basic в операционной системе Linux), который распространяется по лицензии компании AltLinux [15].

При изучении данного курса [17] учащиеся смогут объяснить структуру основных алгоритмических конструкций и использовать их для построения

алгоритмов; определить основные типы данных и операторы; разработать и записать на языке программирования типовые алгоритмы; создавать проекты с использованием визуального объектно-ориентированного программирования.

Объектно-ориентированный подход к решению задач позволяет сформировать у учащихся объектный стиль мышления и способствует подготовке учащихся к дальнейшему изучению программирования.

**УМК «Информатика и ИКТ», автор Макарова Н.В.  
(7) 8-9 классы, основная школа**

Состав УМК:

- учебник «Информатика и ИКТ». 8-9 классы;
- Информатика и ИКТ. Практикум. 8-9 класс;
- Информатика и ИКТ. Задачник по моделированию. 9-11 класс;
- авторская мастерская Макаровой Н.В. [5].

В данном курсе основы программирования рассматриваются в среде Лого. Изучая тему «Программирование» в данной среде, учащиеся знакомятся [18]:

- с программами для реализации типовых конструкций алгоритмов;
- с понятиями процедуры и модуля, процедуры с параметрами;
- с функциями;
- с инструментами логики при разработке программ.

Методика разработки простейших программ в среде ЛогоМиры позволяет развить у обучающихся навыки решения задач с применением алгоритмического, системного и объектно-ориентированного подходов к решению задач; формирует алгоритмическое и логическое мышление; способствует развитию интереса школьников к обучению [9].

**УМК «Информатика и ИКТ», автор Гейн А.Г.  
7-9 классы, основная школа**

Состав УМК:

- рабочие программы;
- учебник;
- рабочая тетрадь;
- задачник-практикум;
- тематические тесты;
- книга для учителя.

В основе курса лежит установка на формирование у учащихся системы базовых понятий информатики и представлений об информационных технологиях, а также выработка умений применять их для решения жизненных задач.

Изучение основ алгоритмизации осуществляется с использованием учебного исполнителя Паркетчик [1], который является свободно распространяемым продуктом. Использование Паркетчика позволяет реализовать принцип наглядности, доступности, переход от простого к сложному.

Русский интерфейс и автоматизированный ввод команд позволяет сосредоточить внимание обучающихся на изучении алгоритмизации, не теряя времени на освоение сложного синтаксиса языка программирования.

Изучение данного курса дает возможность учащимся:

- составлять и записывать алгоритмы для учебных исполнителей с использованием соответствующих алгоритмических конструкций;
- составлять протоколы исполнения алгоритмов;
- распознавать необходимость применения той или иной алгоритмической конструкции при решении задачи;
- использовать готовые вспомогательные алгоритмы при создании нового алгоритма.

Результат анализа учебно-методических комплексов по дисциплине «Информатика и ИКТ» представлен в Таблице 2.

Таблица 2.

Результат анализа УМК по дисциплине «Информатика и ИКТ»

<i>Авторы УМК</i>	<i>Структура УМК</i>	<i>Цель обучения</i>	<i>Среда для изучения алгоритмизации</i>	<i>Особенности УМК</i>
Макарова Н.В.	Учебник 8-9 класс; практикум 8-9 класс; пособия для учителя	Развитие системного мышления и формирование элементов информационной культуры	ЛОГО	Моделирование различных объектов средствами ИТ
Семакин И.Г.	Учебники 8-9 класс; задачник 7-11 класс; пособие для учителя	Обеспечить освоение стандарта	ГРИС PASCAL	После каждой главы система основных понятий; два уровня изложения материала
Угринович Н.Д.	Учебники 8-9 класс; пособие для учителя с диском	Обеспечить освоение стандарта	VBAISIC (QBASIC)	BASIC; диск с тестовыми и другими материалами
Гейн А.Г.	Программа 7-9 класс; учебники 7,8,9 классы; рабочие тетради; Метод. реком. для учителя; задачник-практикум	Применять ИТ для решения жизненных задач	Исполнитель ПАРКЕТЧИК	Направлен на формирование УУД

На основе проанализированных учебных комплексов можно сделать вывод, что основным критерием выбора курса для изучения алгоритмизации и программирования является не только содержание и методические приемы изучения, а также наглядность и простота изучения.

Правильно организованное обучение должно способствовать развитию алгоритмического и логического мышления в естественной для этого обстановке. Такая обстановка должна предоставлять опыт работы с различными моделями; знакомить с общими принципами и методами

программирования; позволять учащимся адаптировать приобретенные навыки для решения жизненных задач.

Для развития алгоритмического мышления обучающегося, а также для реализации его творческих способностей, необходимо создать ему соответствующие условия и предоставить возможность участвовать в групповой деятельности.

Рассмотренные курсы ограничиваются виртуальными средами для исполнителя алгоритмов, что не позволяет в полной степени реализовать принцип наглядности. Также не все УМК предполагают решение большого количества задач, что может привести к усвоению основ алгоритмизации и программирования на недостаточно высоком уровне.

Начинать изучение темы «Алгоритмизация и программирование» целесообразно со знакомства с исполнителями с обратной связью. Такое обучение обеспечит подготовку к последующему изучению объектно-ориентированных языков и языков программирования высокого уровня в наглядной форме. Рассмотрение базовых алгоритмических конструкций, в применении с программируемым устройством, позволит сформировать навыки их использования при решении более сложных практических задач. В сравнении с исполнителями виртуальными, реальные исполнители послужат фундаментом для формирования умения проанализировать задачу и формализовать ее условие применительно к условиям реальной обстановки.

### ***1.3. Обзор существующих исполнителей с обратной связью и методика их использования***

Исполнители с обратной связью используются в качестве методического средства, позволяющего ввести основные понятия алгоритмизации, и начать решать задачи в наглядной среде, освобождённой от излишних трудностей.

Обратная связь – это информация об окружающей обстановке, которую исполнитель использует для выбора нужного варианта выполнения алгоритма.



Действие обратной связи можно описать следующей схемой (см. Рисунок 1):



*Рисунок 1. Действие обратной связи*

Обратная связь предоставляет возможность контролировать результаты выполняемых действий исполнителя во время его работы и следить за изменениями в обстановке.

## **Виртуальные исполнители алгоритмов**

### *ЛОГО-Черепашка*

Исторически первым педагогическим программным средством, предназначенным для обучения детей алгоритмизации, был язык программирования Лого, разработанный в конце 1960-х годов американским педагогом-психологом Сеймуром Пейпертом [10].

В состав Лого входит исполнитель Черепашка, предназначенный для изображения на экране чертежей и рисунков. Последние в свою очередь состоят из прямолинейных отрезков. Программы управления Черепашкой состояются из простых команд: вперед, назад, направо, налево, поднять хвост, опустить хвост. Имеется в виду, что черепашка рисует хвостом, и если хвост опущен, то при перемещении проводится линия, а когда хвост поднят, то линия не рисуется.

В целом язык Лого предназначен для обучения структурной методике программирования, так как в нем имеются все основные структурные команды.

Главное методическое достоинство исполнителя Черепашка – это ясность для ученика решаемых задач и наглядность процесса работы в ходе выполнения программы.

Целью использование Лого может являться развитие алгоритмического и логического мышления, а также творческого потенциала учащихся. Широко используется в УМК Макаровой Н.В. [6].

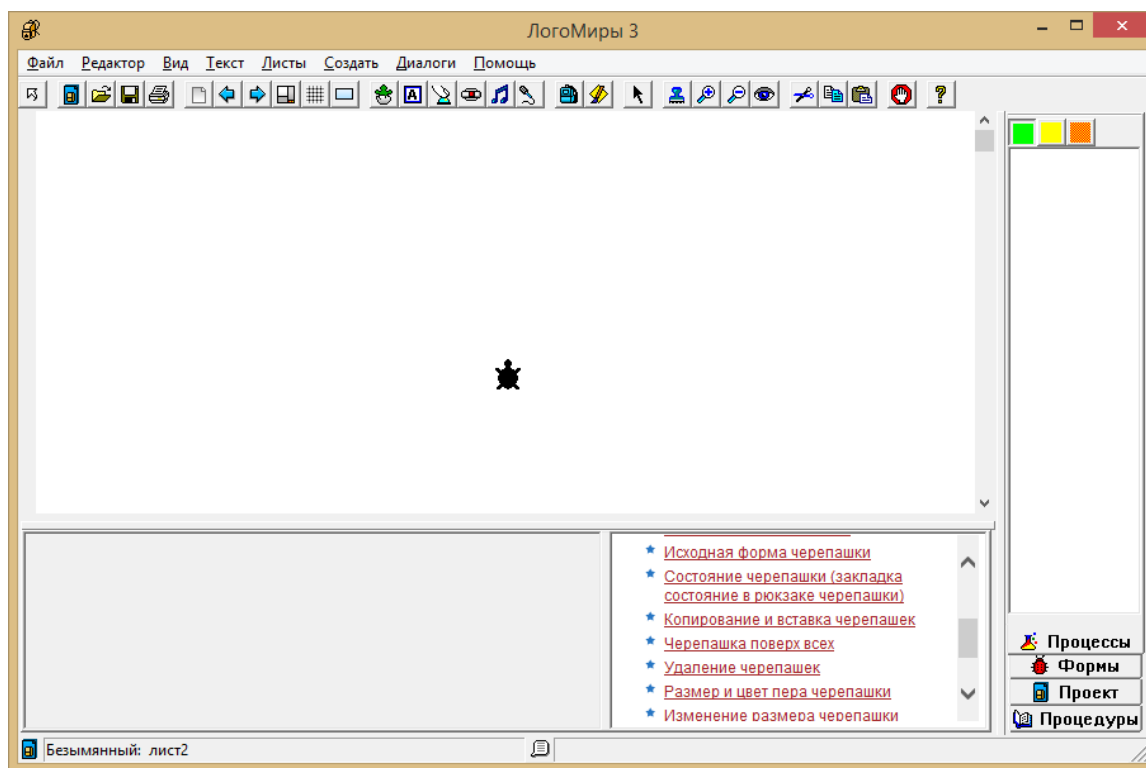


Рисунок 2. Интерфейс программной среды «ЛогоМиры»

В настоящее время существует целый мир диалектов Лого, различающихся набором исходных процедур, наличием и составом встроенных библиотек и интерфейсом среды – LogoWriter, MicroWorlds (ЛогоМиры), ImagineLogo, StarLogo, UCBLLogo, MSWLogo, ComeniusLogo, TerrapinLogo, LogoPlus, KTurtle и другие.

### *Комплект Учебных Миров*

Комплект Учебных МИРов – это система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе.

В системе КуМир используется школьный алгоритмический язык с русской лексикой и встроенными исполнителями Робот и Чертёжник.

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки.

Система команд исполнителя состоит из простых и логических команд: вверх, вниз, влево, вправо, закрасить, сверху свободно, снизу свободно, слева свободно, справа свободно.

В задачах чаще всего требуется заставить исполнителя совершить те или иные действия, например, Роботу может быть дано задание пройти лабиринт, обходя препятствия и закрашивая по пути, отмеченные заранее клетки.

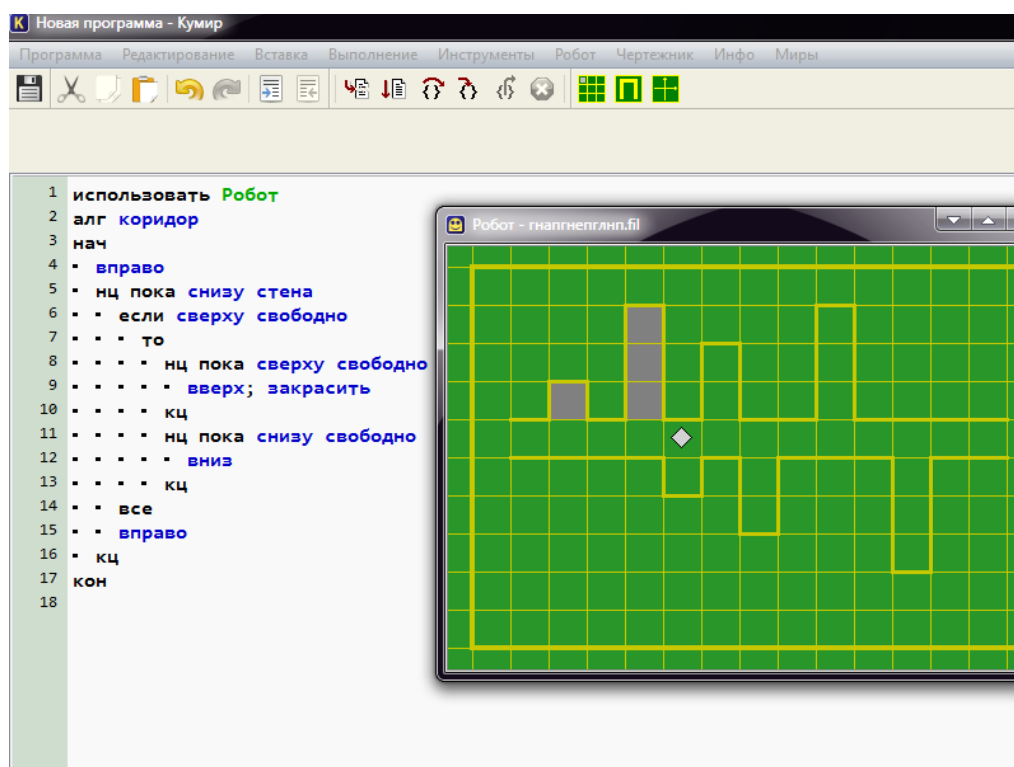


Рисунок 3. Интерфейс программной среды исполнителя Робот

Исполнитель Чертежник предназначен для построения рисунков на координатной плоскости. Чертежник имеет перо, которое можно поднимать, опускать и перемещать.

Система команд исполнителя Чертежник включает шесть команд: опустить перо, поднять перо, сместиться в точку, сместиться на вектор, установить цвет, надпись.

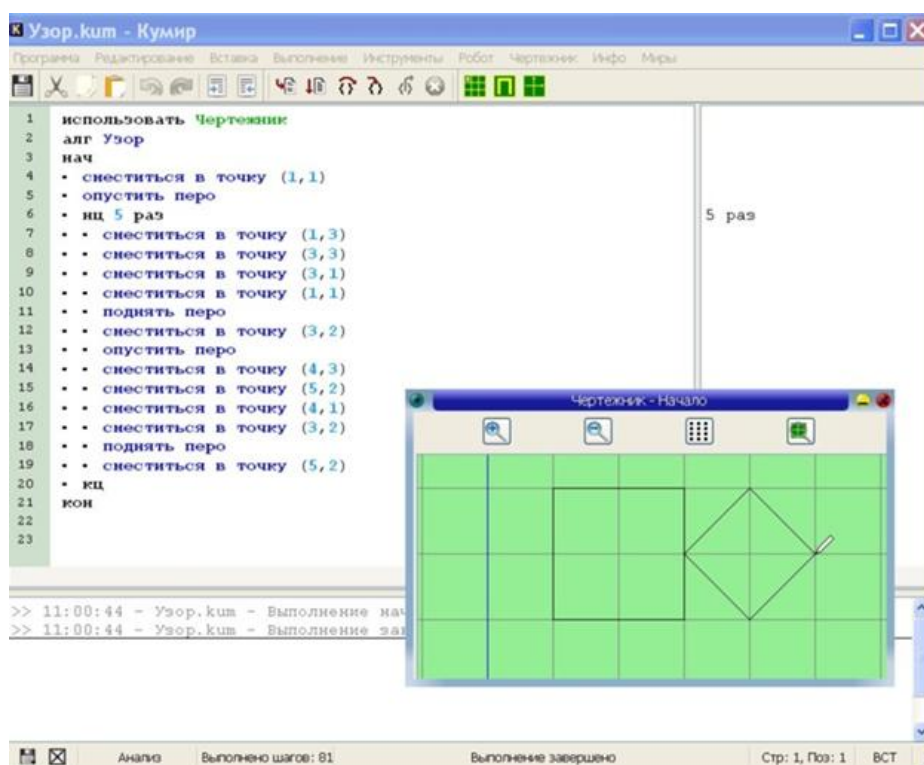


Рисунок 4. Интерфейс программной среды исполнителя Чертёжник

Система КуМир разработана таким образом, что ученик, решая задачи, концентрируется на преодолении соответствующей алгоритмической сложности, а не на освоении особенностей программного обеспечения. Это касается как ввода конструкций языка, так и диагностики ошибок при вводе алгоритма, и средств исполнения, и показа результатов работы написанного алгоритма. В процессе исполнения на экране изображаются и алгоритм, и результаты его работы. Выполнение можно прервать, алгоритм подправить и запустить вновь.

Данная система используется в учебниках первого поколения Кушниренко А.Г. [8].

### Паркетчик

Для того чтобы программировать на языке Паркетчик, предназначена специальная программа «Паркетчик». Среда Паркетчика – это лист бумаги в клетку. Перемещаясь по клеткам, Паркетчик выкладывает квадратные паркетные плитки разного цвета.

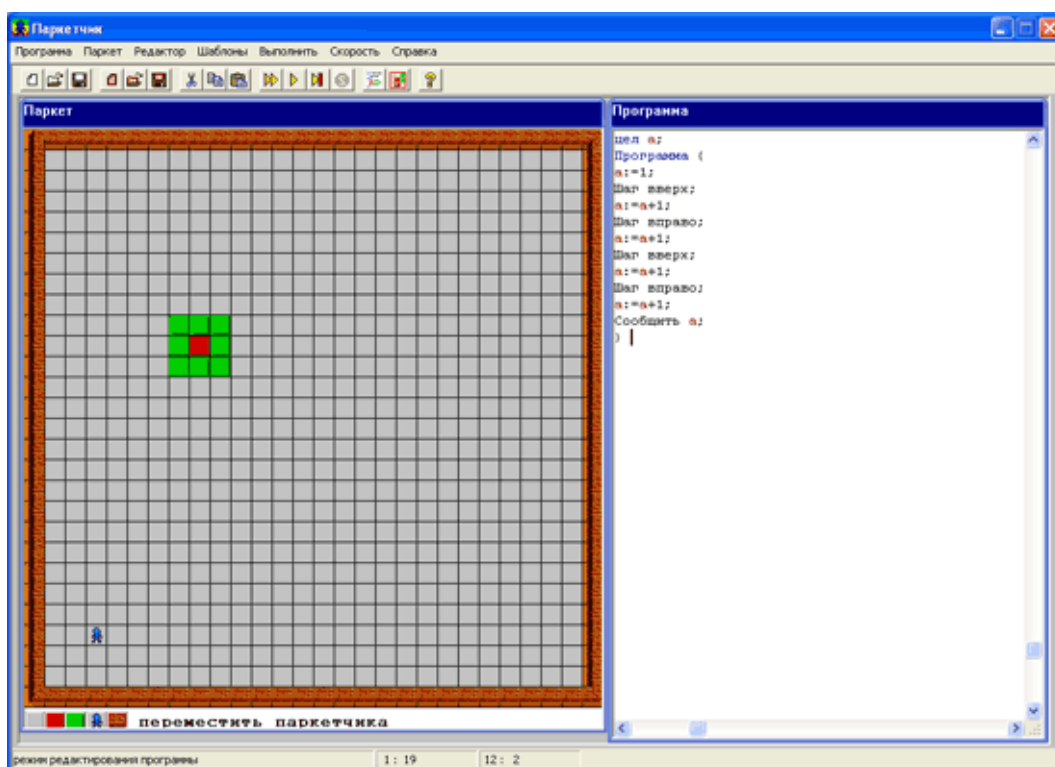


Рисунок 5. Интерфейс программной среды «Паркетчик»

Паркетчик умеет перемещаться по клеточному полю, распознавать цвет плитки, убирать и ставить плитки, распознавать наличие стен в соседних клетках, выполнять несложные математические действия с целыми числами.

Программа «Паркетчик» будет полезна при изучении основ алгоритмизации, так как она способствует не только эффективному освоению приемов алгоритмизации с использованием переменных, но и знакомству с различными структурами данных. Данная программа используется при изучении алгоритмизации в УМК Гейна А.Г. [1].

### Scratch

Scratch – это визуальная объектно-ориентированная среда программирования для обучения школьников младших и средних классов. Scratch был создан как продолжение идей языка Лого и конструктора Лего.

Основными компонентами Scratch-программы являются объекты-спрайты. Спрайт состоит из графического представления – набора кадров-костюмов и сценария-скрипта. Для редактирования костюмов спрайтов в

скретч встроен графический редактор. Действие Scratch-программы происходит на сцене с центром координат в середине сцены.

Для программирования сценариев в Scratch используется подход «drag-and-drop» – блоки из палитры блоков перетаскиваются в область скриптов. По функциональному назначению блоки делятся на 8 групп, принадлежность блока к той или иной группе обозначается его цветом.

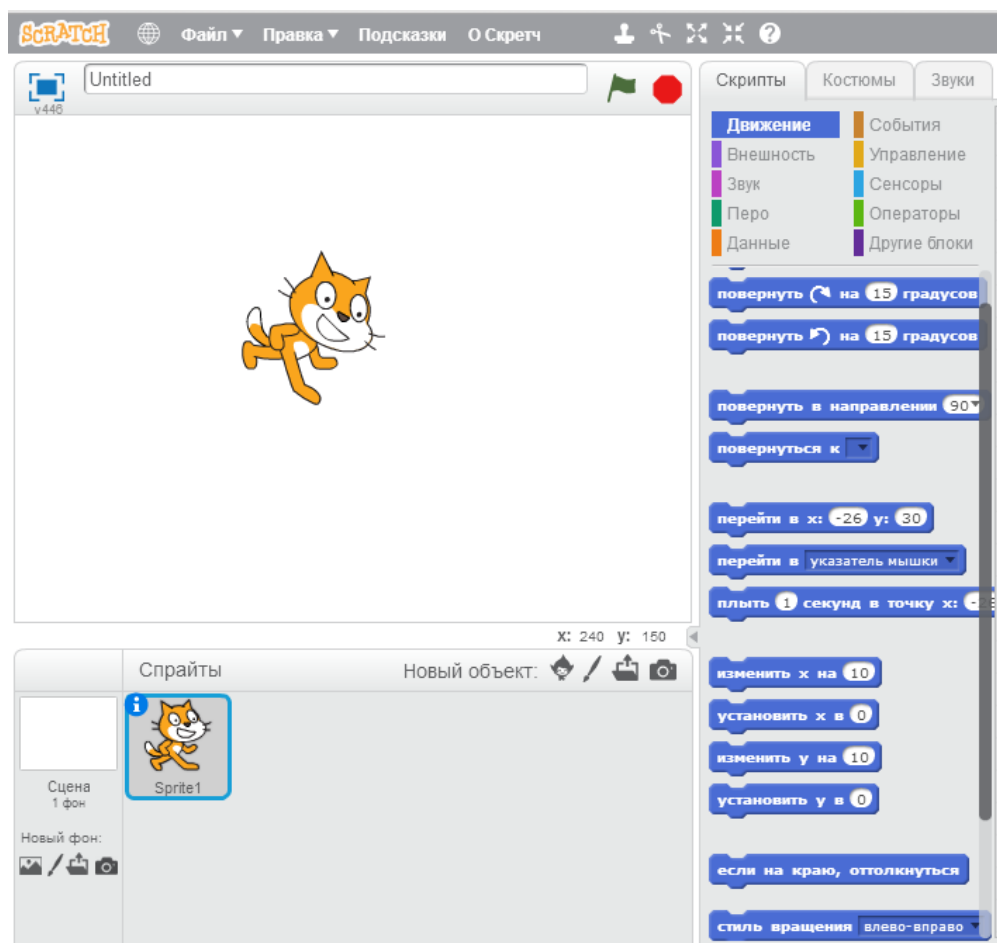


Рисунок 6. Интерфейс программной среды «Scratch»

Данная программа дает возможность учащимся создавать анимированные истории, фильмы, игры и многое другое. Идеология данного программного обеспечения заключается в использовании современных средств, методик и технологий обучения, таких как проблемный подход и проектный метод.

По завершению изучения основных конструкций языка и возможностей программы перед учащимися возможна постановка задачи по разработке и созданию проекта. В качестве проекта могут выступать различные истории,

тематику которых учитель предлагает с учётом возрастных особенностей учащихся [4].

Данная технология, а также разнообразные средства работы с мультимедийными ресурсами обучения стимулируют учащихся к дальнейшему освоению языков программирования и к изучению дисциплины «Информатика и ИКТ» в целом.

Наряду с методической целесообразностью использования виртуальных исполнителей при обучении основам алгоритмизации, имеется и ряд их недостатков. Одним из главных является использование искусственной компьютерной среды. В сравнении с необходимым в реальной жизни, такие исполнители имеют очень ограниченный функционал.

### **Реальные исполнители с обратной связью**

На смену примитивным исполнителям пришли реальные исполнители с обратной связью – робототехнические устройства. У учащихся появилась возможность управлять физическим устройством, с которым можно взаимодействовать. Этому программируемому устройству можно давать различные задания, которые имеют под собой реальное жизненное обоснование.

Явное преимущество реальных исполнителей с обратной связью заключается в использовании датчиков, что делает функционал робототехнического устройства значительно шире компьютерных исполнителей алгоритмов. С одной стороны, это повышает интерес к их применению, а с другой – изучение алгоритмизации становится более полноценным и разносторонним: программируемое устройство может отслеживать состояние элементов окружающей среды и соответственно реагировать.

Одним из ключевых вопросов при реализации методики с использованием реальных исполнителей является подбор конструктора, позволяющего решать современные образовательные задачи.

### *FISCHERTECHNIK*

FISCHERTECHNIK – это пластмассовый развивающий конструктор для детей, подростков и студентов, изобретенный профессором Артуром Фишером в 1964 году [25].

Основным элементом конструктора является блок с пазами и выступом. Такая форма дает возможность соединять элементы практически в любых комбинациях. Также в комплект конструктора входят программируемые контроллеры, двигатели, различные датчики и блоки питания, что позволяет приводить механические конструкции в движение, создавать роботов и программировать их с помощью компьютера.

Для управления моделями, собранными из конструкторов FISCHERTECHNIK, используется программное обеспечение ROBO Pro Software. Программы в нем составляются на графическом языке в виде блок-схем.

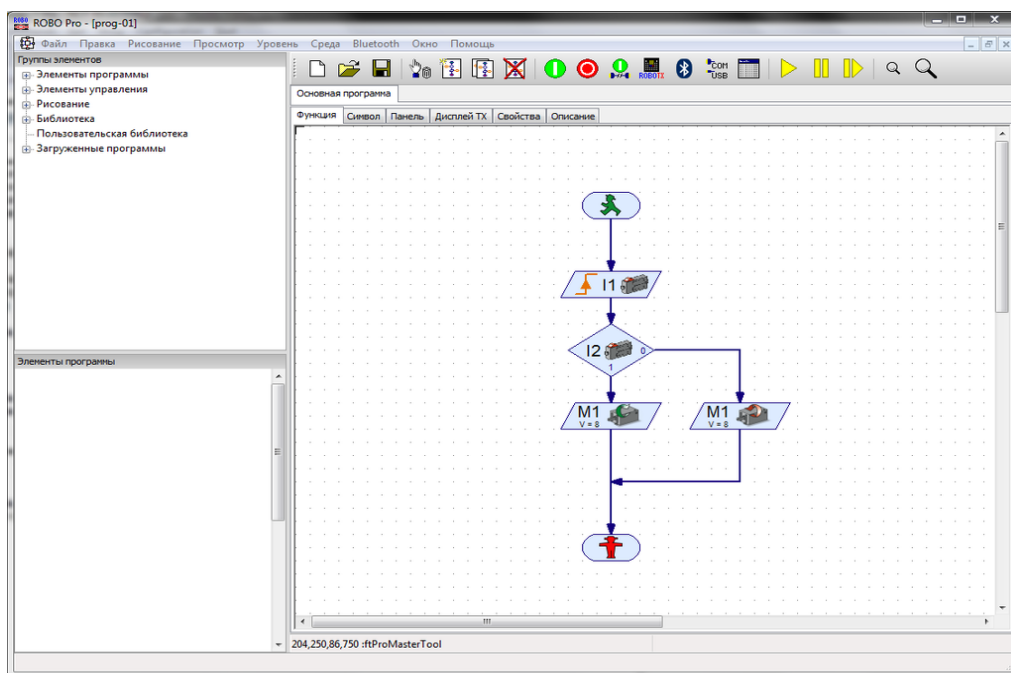


Рисунок 7. Интерфейс программной среды «ROBO Pro»



В среде ROBO Pro существует два режима выполнения управляющих программ, разработанных пользователем – запуск программы на компьютере и запуск программы на контроллере ROBO TX. Эти режимы называются «Онлайн» и «Загрузка в контроллер» соответственно.

В режиме «Онлайн» управляющая программа выполняется на том же компьютере где и среда разработки ROBO Pro. Взаимодействие управляющей программы с входами и выходами контроллера ROBO TX происходит через интерфейс USB или по беспроводной линии связи Bluetooth. Для работы в этом режиме требуется постоянное соединение с компьютером.

В режиме «Загрузка в контроллер» управляющая программа, предварительно записанная в память контроллера, выполняется в контроллере ROBO TX и взаимодействует с входами и выходами через интерфейс встроенной операционной системы реального времени.

### *Robotis Bioloid*

Robotis Bioloid – это набор для создания робота, производимый корейской фирмой Robotis . Данный набор предназначен для образовательных целей. Набор Bioloid включает в себя небольшие сервоприводы, называемые Dynamixel и представляющие собой самостоятельные модули, с помощью которых могут быть собраны роботы различной конструкции, например колёсные или шагающие роботы [24].

В комплект Bioloid входят сервоприводы, набор сенсоров, программное обеспечение, включающее в себя среду 3D-моделирования – RoboMotion и среду программирования на C-подобном языке –RoboTask.

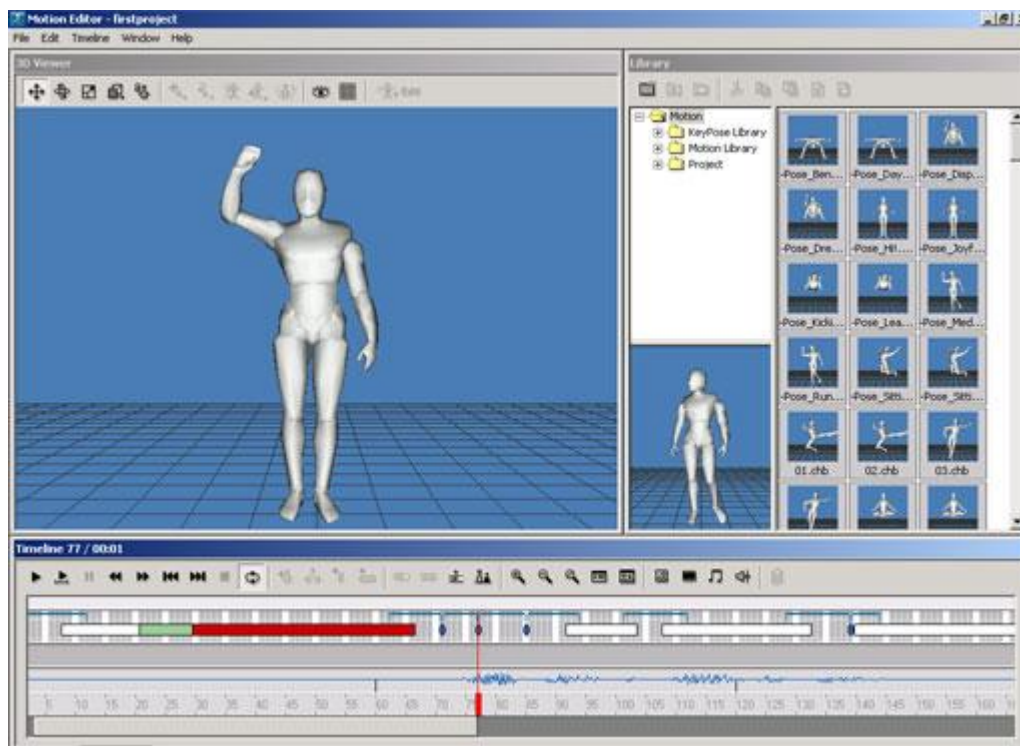


Рисунок 8. Интерфейс программной среды для Robotis Bioloid

Программирование в среде RoboTask более трудоемкий способ управления робототехническим устройством, но оно будет отличной базовой подготовкой для будущего учащихся, так как производится в среде программирования на С-подобном языке.

### *LEGO MINDSTORMS*

Среди имеющихся сегодня на рынке образовательных конструкторов бесспорным лидером, с нашей точки зрения, являются конструкторы, разработанные датской фирмой LEGO [21].

Серия LEGO MINDSTORMS, как робототехнический конструктор впервые была представлена в 1998 году. Стандартные наборы содержали программное и аппаратное обеспечение для создания модифицируемых механических программируемых роботов, так же включали в себя интеллектуальный компьютер, так называемый «кирпич», который управляет системой, ряд модульных датчиков и двигателей, и детали из линейки LEGO Technics для создания механических систем. Первую версию конструктора комплектовали микрокомпьютером RCX.

В 2006 году вышла вторая версия конструктора – NXT.

Базовый набор LEGO MINDSTORMS Education NXT позволяет учащимся собирать и программировать модели реальных роботов.

Учебные цели [7]:

- развитие навыков поиска и выбора оптимального решения задачи, сборки, тестирования и оценки моделей;
- обучение в работе в команде, обмену идеями и навыкам общения;
- приобретение практического опыта работы с датчиками.

В начале 2013 года появился набор конструктора LEGO MINDSTORMS Education Evolution 3.

Данный набор оптимизирован для использования в классе. Он позволяет ученикам создавать, программировать и тестировать свои устройства, используя реальные технологии.

Учебные цели:

- изучение ключевых принципов программирования, развитие алгоритмического мышления, создание и отладка сложных программ по управлению устройствами;
- получение предметных знаний путем практического экспериментирования и моделирования;
- изучение современных технологий с помощью создания моделей, демонстрирующих работу изучаемых концептов;
- подтверждение гипотез опытным путем, включая всесторонний анализ полученных данных, в том числе и прогнозирование;
- получение практического опыта работы с ПК и цифровым измерительным оборудованием;
- развитие навыков общения, умения находить компромиссные решения и работать сообща.

Состав образовательной версии LEGO MINDSTORMS EV3 [22]:

1. Микрокомпьютер EV3 – является «сердцем» и «мозгом» программируемых устройств.
2. Заряжаемый аккумулятор.
3. Большой сервомотор (2 штуки) – создан для работы с микрокомпьютером и имеет встроенный датчик вращения с точностью измерений до 1 градуса. Датчик вращения может использоваться при проведении различных экспериментов для точного считывания данных о расстоянии и скорости.
4. Средний мотор.
5. Датчик касания (2 штуки) – простой, но высокоточный инструмент, который определяет, нажата его кнопка или нет, а также способен определять количество нажатий, как одиночных, так и множественных. Ученики могут использовать его для создания и программирования устройств, способных выйти из лабиринта.
6. Датчик цвета – способен определить восемь различных цветов. Он так же может использоваться как датчик освещенности.
7. Ультразвуковой датчик – генерирует звуковые волны и фиксирует их отражение от объектов, тем самым измеряя расстояние до объектов.
8. Гироскопический датчик – позволяет измерять движение вращения робота, а также улавливать изменения в его движении и положении. С помощью этого датчика легко можно измерить углы.
9. 540 строительных элементов.
10. Набор RJ12 кабелей [23].

С EV3 в комплекте поставляется среда разработки на базе LabView. В данной среде существует такое понятие как проект, который содержит программу для робота, документацию и результаты экспериментов. В проект можно добавлять новые и уже существующие программы.

С точки зрения программирования, можно отметить следующее:

- тесная интеграция между Р-блоком и средой программирования;

- специальная страница с подключенным оборудованием позволяет отслеживать статус Р-блока и получать значения на датчиках в реальном времени;
- возможность работать с массивами;
- возможность досрочного выхода из цикла.

Одной из отличительных особенностей нового набора конструктора LEGO является наличие программы под Android и iOS для управления роботом.

Освоение учащимися LEGO-технологии позволяет:

- дать основные знания в области механики, конструирования и основах автоматического управления;
- обучить их технологии работы в среде программирования для роботов EV3;
- активно принимать участие в соревнованиях и творческих проектах;
- развивать логическое мышление, творческий и познавательный потенциал школьника, его коммуникативные способности с использованием компьютерного инструментария.

При работе с использованием робототехнических конструкторов LEGO MINDSTORMS формулируется умение определять адекватные способы решения учебной задачи на основе заданных алгоритмов, комбинировать известные алгоритмы деятельности в ситуациях, не предполагающих стандартного применения одного из них, мотивированно отказываться от образца деятельности и искать оригинальные решения.

Современный образовательный стандарт по дисциплине «Информатика и ИКТ» выделяет 20% учебного времени на изучении содержательной линии «Алгоритмизация и программирование». Авторы имеющихся учебно-методических комплексов, при изучении данного раздела, отдают предпочтение виртуальным исполнителям алгоритмов.

Согласно ФГОС материально-техническое оснащение образовательного процесса должно обеспечивать возможность проектирования и

конструирования, в том числе моделей с цифровым управлением и обратной связью, с использованием конструкторов.

Разнотипность существующих технических и программных средств, вынуждают нас возвращаться к отбору содержания и методов преподавания курса «Информатика и ИКТ». При выборе из существующих или при создании новых методик обучения необходимо учитывать некоторые общие дидактические принципы:

- принцип научности;
- принцип последовательности и цикличности;
- принцип сознательности усвоения деятельности;
- принцип доступности содержания;
- активность и самостоятельность;
- индивидуализация и коллективность обучения;
- эффективность учебной деятельности;
- связь теории и практики;
- принцип наглядности.

Помимо общих дидактических принципов, стоит отметить, что новые стандарты обучения обладают отличительной особенностью – ориентацией на результаты образования и рассматриваются на основе системно-деятельностного подхода, который призван обеспечить:

- формирование готовности к саморазвитию и непрерывному образованию;
- активную учебно-познавательную деятельность учащихся;
- построение образовательного процесса с учётом индивидуальных возрастных, психологических и физиологических особенностей обучающихся.

Легко реализовать такую обучающую стратегию можно в образовательной среде LEGO MINDSTORMS EV3. Данное программное обеспечение легко в освоении и простое в использовании; включает в себя

интуитивно понятный язык программирования; имеет встроенный редактор контента для учителя и рабочие тетради для учеников.

Организация обучения на базе исполнителей с обратной связью представит процесс изучения основ алгоритмизации и программирования в наглядной форме. Сборка и программирование устройств мотивируют учащихся к выполнению заданий, поставленных на уроке.

На занятиях учащиеся смогут получить организационные навыки в работе, что в свою очередь сформирует фундамент для будущей научно-исследовательской деятельности.

## **Глава 2. Разработка методики обучения программированию с использованием исполнителей с обратной связью**

### ***2.1. Методические рекомендации по использованию исполнителей с обратной связью при изучении раздела «Алгоритмы и исполнители»***

Целью внедрения исполнителей с обратной связью в школьный курс информатики является повышение результативности обучения, интереса к предмету, мотивации к работе на занятиях, а также актуализация ранее полученных знаний из различных предметных областей и умение применять их к реальным жизненным ситуациям.

Использование таких исполнителей способствуют самостоятельному мышлению учащихся, вырабатывает умения находить и решать проблемы, привлекая для этого знания из разных областей. Также, программируя сконструированную модель исполнителя, учащиеся научатся прогнозировать результаты и возможные последствия разных вариантов решения.

Исполнители с обратной связью целесообразно использовать не только на практических занятиях. При объяснении теоретических сведений об основных базовых конструкциях, реальный исполнитель будет являться лучшим наглядным средством обучения. Особенностью использования исполнителей с обратной связью также является и то, что усложнение поставленной задачи может наглядно показать различия между существующими алгоритмическими конструкциями, что, в свою очередь, способствует эффективному овладению навыками программирования.

Проведя анализ существующих исполнителей с обратной связью, мы сделали вывод о том, что действительно универсальным и подходящим для внедрения в любой школьный предмет является конструктор LEGO MINDSTORMS Education. Обучение с LEGO MINDSTORMS можно применять к любому уровню знаний и навыков, возрасту, учебной цели и ситуации.

Разрабатываемые методические рекомендации для обучения основам алгоритмизации и программирования при помощи реальных исполнителей



будут основываться на использовании робототехнического устройства LEGO MINDSTORMS EV3.

Основные задачи:

- развитие творческих способностей и логического мышления обучающихся;
- развитие умений выстраивать гипотезу и сопоставлять ее с полученным результатом;
- развитие умения творчески подходить к решению задачи;
- развитие умения применять знания из различных областей знаний;
- развитие умения излагать мысли в четкой логической последовательности, отстаивать свою точку зрения;
- получение навыков проведения физического эксперимента.

Основные этапы обучения:

- конструирование исполнителей на базе микропроцессора EV3;
- работа в среде программирования MINDSTORMS EV3;
- составление программ управления сконструированным устройством, с использованием базовых алгоритмических конструкций.

Для внедрения в образовательный процесс реальных исполнителей необходимо обеспечить кабинет информатики необходимым оборудованием и программным обеспечением:

1. Компьютер (минимальные требования):

- Windows Vista или более поздние версии Windows;
- процессор – 2 ГГц;
- оперативная память – 2 Гб;
- свободное место на жестком диске – 2Гб;
- разрешение экрана 1024x768
- 1 доступный USB-порт.

2. Программное обеспечение LEGO MINDSTORMS Education EV3.

Бесплатно загрузить программное обеспечение на русском языке можно с сайта [LEGO.com/mindstorms](http://LEGO.com/mindstorms). Также на этом сайте доступно руководство пользователя в формате \*.pdf.

### 3. Проектор.

Подсоединяемый к компьютеру проектор радикально повышает уровень наглядности в работе учителя, предоставляет возможность учащимся демонстрировать результаты своей работы всему классу.

### 4. Управляемые компьютером устройства.

В качестве управляемого устройства используется базовый набор LEGO MINDSTORMS Education EV3.

Использование управляемых устройств предполагает произвести предварительную сборку модели, это значительно сократит затрату времени от урока.

Во время занятия учащиеся могут доукомплектовывать готовые модели различными датчиками, позволяющими внести в урок разнообразие и поддерживать интерес учащихся в решении конструкторских и проектных задач. Конструирование своего неповторимого устройства позволит учащимся отклоняться от инструкций, включая собственную фантазию. Возможность создания оригинальных моделей поспособствует возрастанию мотивации и активности каждого учащегося, что приведет к выводу обучения на новый продуктивный уровень.

Пример сконструированной модели исполнителя представлен в Приложении 1.

Для углубленного знакомства с различными видами соединения деталей и проектированием моделей исполнителя возможно введение элективных курсов. Реализуемой задачей в рамках курса может стать задача выработки навыков читать чертежи и взаимодействовать в команде.

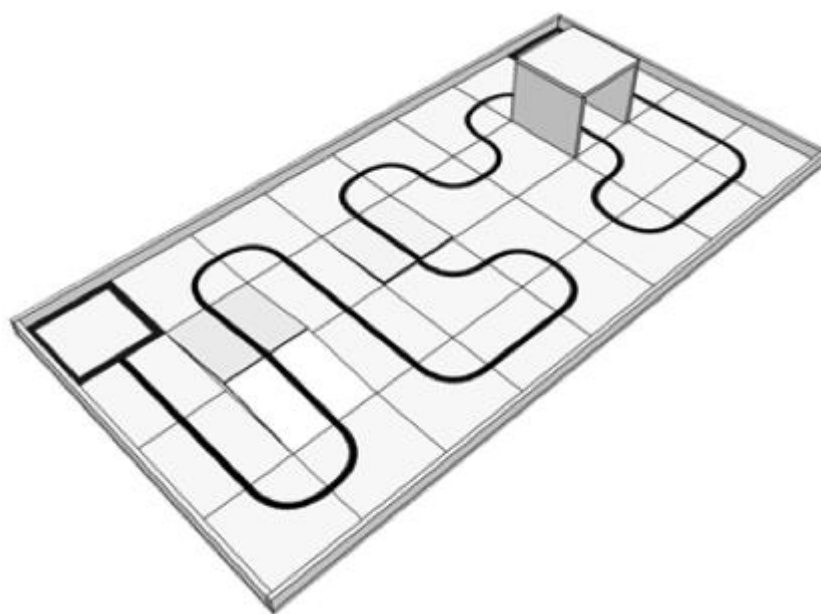
Методика LEGO Education основана на парной работе учащихся, поэтому комплектацию конструкторами рекомендуется осуществлять из расчета один

конструктор на двух учащихся. Это позволяет ученикам приобретать навыки сотрудничества и одновременно справляться с индивидуальными заданиями.

Работа также возможна и при наличии 5-7 комплектов, так как на одно и то же устройство в течение урока могут присылать программы различные группы обучающихся. Вместе с этим основной принцип обучения LEGO-технологиям – «шаг за шагом» – обеспечивает учащимся возможность работать в собственном темпе.

Помимо образовательных наборов программируемых устройств важно оснастить кабинет информатики траекториями для движения исполнителя. Как правило, траектория движения представляет собой черную линию на белом фоне. Чем больше траекторий имеется в образовательном учреждении, тем разнообразнее круг задач, решаемых в рамках дисциплины.

Для безопасной эксплуатации робототехнических устройств рекомендуется работать на траекториях, оснащенных бортиками (см. Рисунок 9). Это снизит риск падения и поломки собранных устройств, что, в свою очередь, поможет избежать непредусмотренных трат времени на конструирование нового исполнителя.

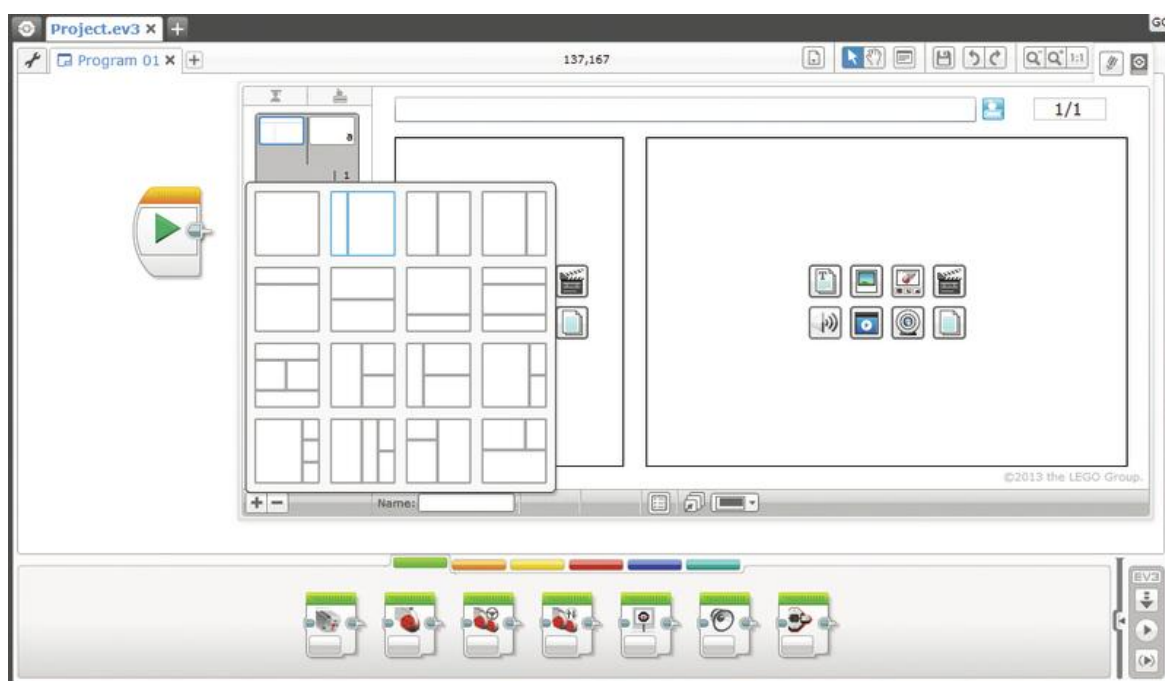


*Рисунок 9. Траектория движения для исполнителя, оснащенная бортиками*

## Рекомендации по использованию программного обеспечения EV3

### *Редактор контента*

Важная особенность программного обеспечения EV3 заключается в том, что весь предоставляемый контент можно редактировать. Редактор контента (см. Рисунок 10) позволяет учителю редактировать стандартные учебные материалы EV3, адаптируя их под цели и задачи учебного курса. Также имеется возможность создавать и добавлять собственные учебные материалы, учитывающие уровень подготовки непосредственно ваших учеников, как в целом класса, так и индивидуальный.



*Рисунок 10. Редактор контента для учителя*

Редактор контента также позволяет учащимся документировать свои успехи и открытия, а именно:

- записывать полные описания их рабочих процессов;
- вставлять свои собственные страницы;
- добавлять картинки и видео их роботов в действии;
- обмениваться своими уникальными проектами с другими учащимися.

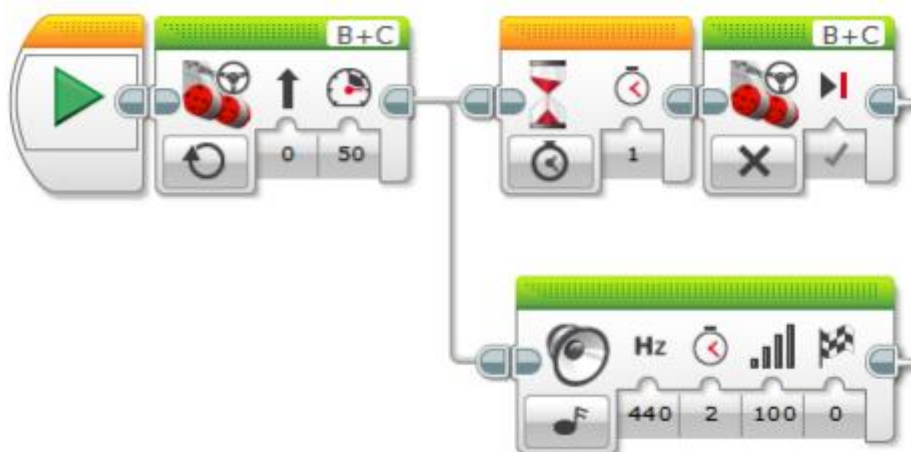
Такие электронные тетради позволят упростить процесс проверки и оценивания учеников. В любой момент времени учитель сможет просмотреть работы учеников, а также оставить им свой комментарий или отзыв.

Оценивая работы при помощи электронной тетради, учитель сможет автоматизировать процесс проставления оценок, создав цифровой класс для себя и для учеников.

### *Программы*

В среде EV3 имеется возможность создавать программы, перетаскивая программные блоки с палитры в область программирования. Когда программные блоки находятся близко друг к другу, они автоматически состыковываются.

При запуске программы программные блоки будут запускаться в том порядке, в котором они отображаются на экране, слева направо. Для длинных программ рекомендуется разбить программу на более мелкие фрагменты (см. Рисунок 11) программных блоков с промежутками между фрагментами. Это позволит сделать программу более простой для понимания. Например, отдельный фрагмент для каждого действия позволит с легкостью следить за программой.



*Рисунок 11. Разбиение программы на фрагменты*

### *Подключение к EV3*

Для установления связи и получения программ модулем EV3 с компьютера имеется три способа:

- подключение через USB;
- подключение через Bluetooth;
- подключение через Wi-Fi.

Для использования подключения через Wi-Fi требуется точка доступа, и маршрутизатор, также, по сравнению с подключением через Bluetooth и USB, такое подключение потребляет больше мощности батареи. Подключение же через Bluetooth невозможно без Bluetooth-модулей, установленных на рабочей станции. Исходя из этого, рекомендуется использовать USB-подключение, так как оно более быстрое и обеспечивает связь с устройством без дополнительных атрибутов соединения.

### **Особенности организации уроков-практик**

Общие вопросы, возникающие при организации уроков-практик с использованием реальных исполнителей:

- расстановка мебели в кабинете;
- вопросы хранения конструкторов.

Мебель в кабинете должна быть расставлена свободно, для того, чтобы учащиеся могли передвигаться, не создавая друг другу неудобств и, тем самым, избегая травмоопасных ситуаций. Необходимо установить столы и регулируемые по высоте стулья для индивидуальной работы учащихся. Также в центре кабинета рекомендуется оборудовать площадку для тренировочных и контрольных заездов.

Пример схемы расположения учебной мебели в кабинете информатики и ИКТ представлен в Приложение 2.

Для хранения программируемых комплектов конструкторов лучше использовать встроенные шкафы. Если кабинет оборудован не стационарными

компьютерами, а ноутбуками, то необходимо предусмотреть также стеллажи для их хранения и коробки для хранения зарядных устройств.

Вопрос о безопасности и сохранности всего оборудования остается открытым для каждого учебного заведения.

При организации непосредственно учебного процесса с использованием робототехнических устройств возникает следующий ряд вопросов:

- планирование процедуры урока;
- результаты занятия;
- оценка результатов занятия.

Планирование урока с использованием программируемых устройств, имеет ряд отличительных особенностей. В первую очередь – это значительные затраты времени на сборку устройств. Как выше было указано, эта проблема вполне решаема за счет использования ранее собранных исполнителей.

Вторая особенность связана с тем, что каждая группа учащихся на занятии будет работать в индивидуальном темпе. Часть класса будет организовывать свою деятельность размеренно, решая поставленную задачу, не выходя за рамки урока, оставшаяся часть учащихся может справляться с заданиями гораздо быстрее. В последнем случае перед учителем встает вопрос об инвариантности поставленной задачи. Для поддержания интереса учащихся, как для изучаемого раздела, так и к дисциплине в целом, необходимо иметь в запасе ряд заданий повышенной сложности.

С учетом особенностей использования исполнителей с обратной связью, урок может содержать семь этапов:

**1 этап.** Деление учащихся на рабочие мини-группы.

В связи с ограниченным временем урока, целесообразно использовать приемы, которые позволят быстро произвести деление класса на группы. Например, по выбору заранее подготовленных цветных карточек. Группы могут быть сформированы один раз на вводном занятии или же деление может происходить непосредственно в начале каждого урока.

## **2 этап. Постановка задачи.**

Для того чтобы мотивировать учащихся к работе необходимо подбирать сюжетные задачи. Например, «Одним из самых популярных роботов в настоящее время является робот-пылесос. Основная траектория движения робота – окружность. Представьте, что вы – программист и вам поручено написать программу, которая заставляет робота ехать по окружности».

## **3 этап. Обсуждение способов решения задачи.**

На данном этапе учитель вместе с учениками обсуждает возможные варианты решения задачи. На этом же этапе возможна постановка индивидуальных задач.

**4 этап.** Конструирование робота с необходимыми блоками, моторами и сенсорами.

Учащиеся используют готовые устройства, дополняя их датчиками, необходимыми для решения поставленной задачи.

## **5 этап. Программирование.**

## **6 этап. Отработка на полигоне.**

Учащиеся выполняют тренировочные заезды, размышляют над тем, что можно улучшить или изменить в конструкции робота или программе для более качественного решения поставленной задачи.

## **7 этап. Подведение итогов.**

Итоговый контроль знаний и умений может быть реализован посредством мини-соревнований, где каждая группа учащихся демонстрирует результаты решения поставленной задачи. Здесь может учитываться время и скорость прохождения устройством дистанции; точность выполняемых действий; точность калибровки датчиков и многое другое.

## **Конспект урока-практики**

Практическое применение разработанных методических рекомендаций отражено в примере конспекта урока-практики, направленного на



формирование навыков применения линейных алгоритмических конструкций. При составлении конспекта были учтены все особенности планирования урока с использованием исполнителей с обратной связью.

**Продолжительность урока:** 45 минут.

**Тема урока:** решение задач на составление линейных алгоритмов.

**Класс:** 9.

**Тип урока:** комбинированный.

**Вид урока:** практическая работа.

**Оборудование:** мультимедийная система, секундомеры, наборы конструктора LEGO Mindstorms EV3, ноутбуки, схемы.

**Цели урока:**

- сформировать начальные навыки работы с программным обеспечением EV3;
- ознакомиться с принципами работы датчика освещенности из базового набора LEGO Mindstorms EV3;
- усвоить понятия алгоритм, исполнитель, обратная связь;
- научиться составлять простейшие алгоритмические конструкции в среде программирования EV3;
- систематизировать знания по разделу «Алгоритмы и исполнители».

В результате занятия, обучающиеся должны овладеть следующими универсальными учебными действиями:

Регулятивные:

- систематизировать и обобщить знания по теме «Алгоритмические конструкции: следование» для успешной реализации алгоритма работы собранного робота;
- научиться программировать робототехнические устройства в среде EV3.

Познавательные:

- изучение основ конструирования, знакомство с датчиком освещенности, умение программировать с помощью среды программирования EV3;
- экспериментальное исследование, оценка влияния отдельных факторов на скорость движения робота.

Коммуникативные:

- развить коммуникативные умения при работе в группе или команде.

Личностные:

- развитие памяти и мышления, возможность изучения робототехники в старших классах.

### **Ход урока:**

1. Организационный момент.

*Учитель приветствует учащихся и раздает рабочие материалы.*

Для демонстрации работы исполнителя, понадобится базовая модель устройства из Приложения 1.

2. Постановка задачи.

*Учитель:* наша сегодняшняя задача – запрограммировать устройство, которое будет взаимодействовать с окружающей средой, в частности обнаруживать линию, при помощи распознавания её яркости.

Использование цветных линий является одним из простых способов управления движением программируемых устройств. В поставленной задаче, для того, чтобы научить наших роботов обнаружению линии, мы будем использовать датчик освещённости. Нам понадобится режим измерения отраженного цвета. В этом режиме активируется светоиспускающий элемент. Свет, выпущенный этим элементом, отражается от поверхности и попадает в светочувствительный элемент.

В зависимости от того насколько светлая отражающая поверхность, в светочувствительный элемент приходит разное количество света. Это количество света преобразуется в цифровое значение и передается в

программу. Чем темнее поверхность, тем меньше света – в программу приходят маленькие значения; чем светлее поверхность, тем больше света – программа оперирует с большими значениями.

### 3. Обсуждение способов решения задачи.

*Учитель:* как вы думаете, какие действия должны совершать наши роботы?

*Учащиеся делятся своими предположениями:* например, робот совершит движение по кругу, обнаружит черную линию и остановится.

*Учитель:* я предлагаю вам написать программу, которая позволит нашему устройству двигаться до тех пор, пока датчик света не определит черный цвет, после чего робот должен остановиться.

*Примечание:* в программе используется блок «Ожидание» датчика цвета; «Сравнение» – режим «Цвет» для тестирования черного цвета.

### 4. Программирование.

*Учащиеся составляют первый вариант программы.*

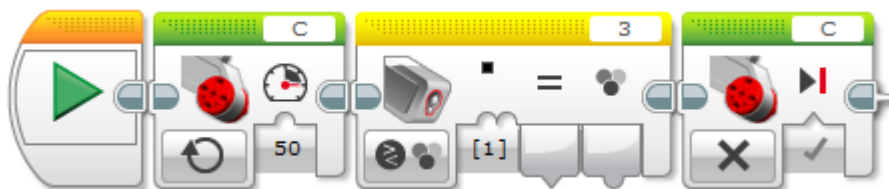


Рисунок 12. Движение до черной линии (вариант 1)

*Учитель:* теперь давайте попробуем усовершенствовать нашу программу, используя блок «Ожидание» датчика цвета – «Сравнение» – режим «Яркость отраженного света». Наши устройства должны ожидать, пока яркость света не достигнет менее 50 %.

*Учащиеся составляют второй вариант программы. Опытным путем, они должны прийти к тому, что для получения наилучших результатов нужно установить порог чувствительности датчика.*



Рисунок 13. Движение до черной линии (вариант 2)

*Учитель:* теперь измените программу таким образом, чтобы робот двигался вперёд до тех пор, пока, не обнаружит линию.

*Примечание:* перед загрузкой программы сохраните её под новым именем.



Рисунок 14. Движение до черной линии (вариант 3)

*Ученикам нужно внести изменения в программу, заменив блоки управления большим мотором блоками – «Рулевое управление». Робот выполнит движение вперед доедет до черты и остановится.*

## 5. Отработка на полигоне.

*Учитель:* а теперь давайте поэкспериментируем, может ли ваше устройство обнаружить более светлые линии. Усовершенствуйте программу так, чтобы она могла обеспечить обнаружение датчиком цвета линии более светлого тона.

*При обнаружении линии более светлого тона необходимо увеличить порог чувствительности датчика цвета, подобрать его нужно опытным путем.*

*Примечание:* перед загрузкой программы сохраните её под новым именем.

## 6. Подведение итогов.

Для подведения итогов, можно расширить рамки задания, предложив ученикам:

- дать пояснения к своим программам и объяснить функции каждого блока с использованием инструмента «Комментарий»;
- поэкспериментировать с программированием обнаружения линии других цветов (например, красного, синего и белого);
- запрограммировать движение вперед до второй или третьей обнаруженной линии.

### **Элементы использования исполнителей с обратной связью при обучении программированию**

Для повышения эффективности обучения необходимо грамотно сочетать используемые технические средства обучения с применением следующих методов:

- объяснительно-иллюстративный метод – предъявление информации ученикам различными словесными и наглядными способами;
- эвристический метод – метод творческой деятельности;
- проблемный метод – постановка проблемы учителем и самостоятельный поиск её решения учащимися;
- программированный метод – установка набора операций, которые необходимо совершить учащимся в ходе выполнения практических работ;
- репродуктивный метод – воспроизводство знаний и способов деятельности по примеру учителя;
- частично-поисковый метод – решение проблемных задач с помощью учителя;
- поисковый метод – самостоятельное решение проблем учащимися;
- метод проблемного изложения – постановка проблемы педагогом, решение ее самим педагогом, соучастие обучающихся при решении.

В рамках содержательной линии «Алгоритмы и исполнители» робототехнические комплексы LEGO MINDSTORMS могут применяться по следующим направлениям:

1. Демонстрация.

Для демонстрации применимы объяснительно-иллюстративный и репродуктивный методы. Например, при введении понятий управления, обратной связи и основных алгоритмических конструкций, учитель может наглядно демонстрировать поведение заранее запрограммированного устройства или уже на самых первых уроках включать в деятельность учеников, предлагая написать уже составленный алгоритм действий для исполнителя. Это обеспечит лучшее усвоение понятий о новых процессах в сознании учащихся. Параллели, проведенные между понятиями алгоритмизации и реальной жизненной обстановкой, станут фундаментом к дальнейшему обучению основ программирования на языках высоко уровня.

2. Фронтальные лабораторные работы и опыты.

Практические занятия остаются наиболее значимыми в процессе формирования алгоритмического мышления учащихся. Организация уроков-практик с использованием исполнителей с обратной связью строится на парной работе учащихся. В таком обучении основное взаимодействие происходит между двумя учениками, иными словами, реализуются процессы взаимообучения и взаимоконтроля. Стоит заметить, что в подростковом возрасте слово и помощь товарища оказывается полезнее помощи учителя. Также в подобных мини-группах происходит более интенсивный обмен информацией и учащиеся, различного уровня подготовки и мотивации, успешнее осваивают новый материал, за счет этого обучение становится более продуктивным.

3. Исследовательская проектная деятельность.

Проектная деятельность является основным компонентом для развития творческих способностей учащихся. Работа над проектом формирует у

учащихся образ цельного знания. Задачи на разработку уникального устройства повышают мотивацию учащихся к самостоятельному получению дополнительных знаний, формируют научное познание.

Элементы использования исполнителей с обратной связью при обучении программированию представлены в Таблице 3.

Методика обучения программированию с использованием исполнителей с обратной связью предполагает использование компьютеров совместно с конструкторами. Важно отметить, что компьютер используется как средство управления моделью, то есть его использование направлено на составление управляющих алгоритмов для собранных моделей. Учащиеся получают представление об особенностях составления программ управления, автоматизации механизмов, моделировании работы систем. Методические особенности реализации предполагают сочетание возможности развития индивидуальных творческих способностей и формирование умений взаимодействовать в коллективе.

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения обучающихся практических заданий. Итоговый контроль реализуется в форме соревнований.

Таблица 3.

**Элементы использования исполнителей с обратной связью при обучении программированию**

	<i>Тема, соответствующая разделу «Алгоритмы и исполнители»</i>	<i>Элементы использования реальных исполнителей</i>	<i>Примечание</i>
1	Алгоритм. Свойства алгоритма. Способы записи алгоритмов; блок-схемы. Возможность автоматизации деятельности человека.	Демонстрация выполнения нескольких алгоритмов. Например, движение вдоль линии.	Теоретическое занятие
2	Исполнители алгоритмов (назначение, среда, режим работы, система команд). Компьютер как формальный исполнитель алгоритмов (программ).	Демонстрация возможностей робототехнического устройства как исполнителя алгоритма. Обсуждение возможных вариантов использования исполнителя на практике. Демонстрация формального исполнения программы.	Теоретическое занятие
3	Алгоритмические конструкции: следование, ветвление, повторение. Разбиение задачи на подзадачи, вспомогательный алгоритм.	Демонстрация поведения исполнителя, использующего в своей работе следование, ветвление, повторение, подпрограмму.	Теоретическое занятие
4	Алгоритмы работы с величинами: типы данных, ввод и вывод данных.	Блок «Переменная» и блок «Константа». Считывание и запись значений. Демонстрация поведения исполнителя с алгоритмом, содержащим обработку величин.	Теоретическое занятие
5	Языки программирования, их классификация.	Язык программирования EV3.	Теоретическое занятие
6	Правила представления данных.	Представление данных на языке программирования EV3.	Теоретическое занятие
7	Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла. Правила записи программы.	Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла на языке программирования EV3.	Теоретическое занятие
8	Этапы разработки программы: алгоритмизация – кодирование – отладка – тестирование.	Блок «Логика». Демонстрация поведения исполнителя и алгоритма с использованием блока «Логика».	Теоретическое занятие
9	Обрабатываемые объекты: цепочки символов, числа, списки, деревья.		Теоретическое занятие
10	Разработка линейного алгоритма (программы).	Движение исполнителя по заранее заданной траектории. Реакции исполнителя на звук, расстояние, касание.	Практическое занятие
11	Разработка алгоритма (программы), содержащей оператор ветвления.	Выбор исполнителем модели поведения в зависимости от состояния окружающей среды.	Практическое занятие



		Знакомство с комплектом датчиков.	
12	Разработка алгоритма (программы), содержащей оператор ветвления.	Выбор роботом модели поведения в зависимости от состояния окружающей среды.	Практическое занятие
13	Разработка алгоритма (программы), содержащей оператор цикла.	Демонстрация и разработка циклического алгоритма движения исполнителя. Например, рисование «восьмерки» при движении. Сравнение длины линейного и циклического алгоритма.	Практическое занятие
14	Разработка алгоритма (программы), содержащей оператор цикла.	Программирование исполнителя в среде EV3, используя внутренние команды.	Практическое занятие
15	Алгоритмические конструкции: следование, ветвление, повторение.	Составление вместе с учащимися алгоритмов движения исполнителя с ветвлением и повторением.	Практическое занятие
16	Разбиение задачи на подзадачи, вспомогательный алгоритм. Разработка алгоритма (программы), содержащей подпрограмму.	Блок как вспомогательный алгоритм. Создание программы движения исполнителя по спирали.	Практическое занятие
17	Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла. Правила записи программы. Разработка алгоритма (программы) по обработке одномерного массива.	Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла на языке программирования EV3.	Практическое занятие
18	Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла. Правила записи программы. Разработка алгоритма (программы) по обработке одномерного массива.	Обработка массивов данных роботом. Демонстрация режима сборки и обработки данных исполнителем.	Практическое занятие
19	Создание алгоритма (программы), решающего поставленную задачу.	Сконструировать и запрограммировать устройство, решающее свою, неповторимую задачу.	Практическое занятие

Опираясь на ФГОС важно выделить универсальные учебные действия, формируемые в результате внедрения исполнителей с обратной связью в образовательный процесс:

- мотивационная основа деятельности;
- умение планировать своё действие в соответствии с поставленной задачей и условиями её реализации;

- оценивать правильность выполнения действия;
- осуществлять анализ объекта с выделением существенных признаков и несущественных;
- осуществлять синтез как составление целого из частей;
- допускать возможность существования у людей различных точек зрения, в том числе не совпадающих с его собственной, ориентироваться на позицию партнёра в общении и взаимодействии;
- договариваться и приходить к общему решению совместной деятельности.

## ***2.2. Содержание уроков-практик с использованием исполнителей с обратной связью***

### **Практическое занятие №1 Разработка линейного алгоритма**

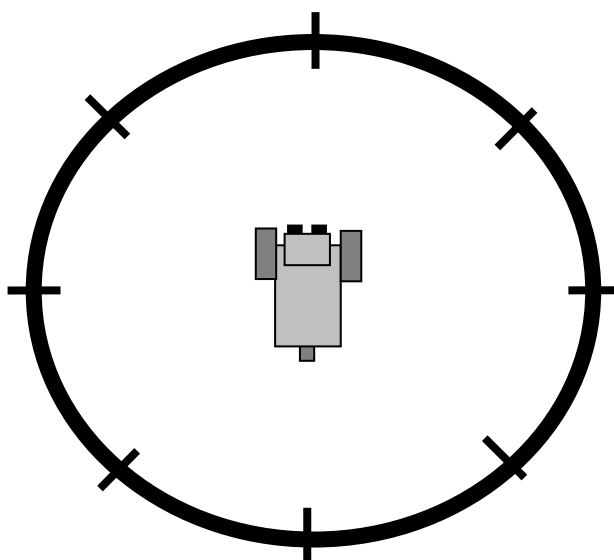
**Пример задания.** Составить программу, заставляющую исполнителя обнаружить черную линию.

**Пример задания.** Исполнитель находится в центре окружности. С помощью коротких отрезков окружность разделена на восемь равных частей (см. Рисунок 15).

Ответить на вопрос – на сколько градусов должен повернуться вал левого двигателя, чтобы исполнитель повернулся вправо на угол в:

- а) 45 градусов;
- б) 90 градусов;
- с) 180 градусов?

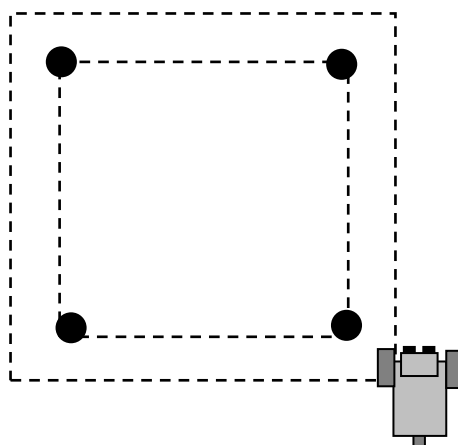
Провести экспериментальную проверку, написав программы поворота исполнителя на указанные углы.



*Рисунок 15. Игровое поле – окружность*

**Пример задания.** На игровом поле в вершинах воображаемого квадрата стоят флажки, сделанные из деталей LEGO-конструктора. Исполнитель устанавливается учениками самостоятельно, таким образом, чтобы он находился рядом с одним из флажков с внешней стороны квадрата (см. Рисунок 16).

Составить программу движения исполнителя вдоль периметра квадрата, таким образом, чтобы он обогнул все четыре флажка, не задев их, но и не удаляясь от стороны квадрата более чем на 20см. Задание считается выполненным, если исполнитель вернулся в начальную точку движения с погрешностью не более 5-10см. Пересечение воображаемой линии соединяющей вершины квадрата считается недопустимым.



*Рисунок 16. Игровое поле – воображаемый квадрат*

## Практическое занятие №2-3

### Разработка программы, содержащей оператор цикла

**Пример задания.** Составить программу движения исполнителя по заданной траектории с одним датчиком света, используя пропорциональное линейное уравнение.

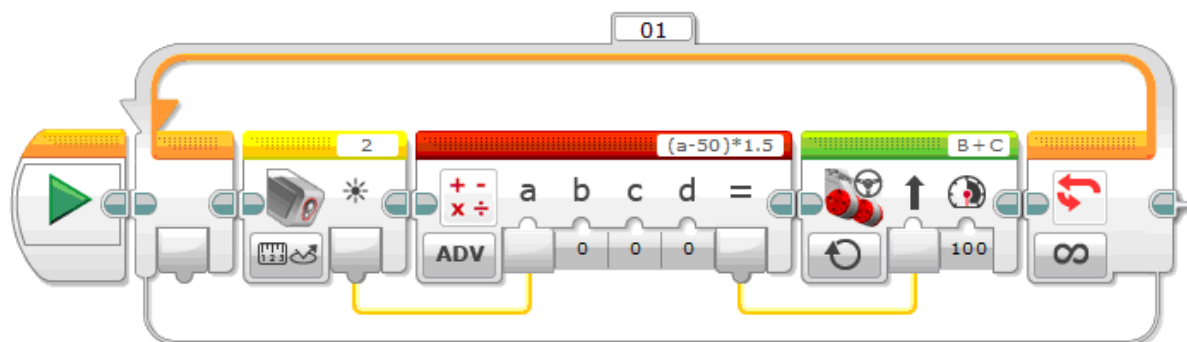


Рисунок 17. Фрагмент программы движение исполнителя по заданной траектории с одним датчиком света

*Методические рекомендации к заданию:* реализуемый алгоритм основывается на том, что возвращаемые датчиком показания градиентно варьируются. Исполнитель сохраняет положение датчика света на границе чёрной линии. Преобразовывая входные данные от датчика света, система управления формирует значение скорости поворота робота. При тренировочном заезде учащиеся увидят, что исполнитель излишне раскачивается и его скорость прохождения траектории очень низкая. На основе опытов учащихся учитель предлагает следующее задание.

**Пример задания.** Составить программу движения исполнителя по заданной траектории с двумя датчиками света, используя пропорциональное линейное уравнение.

**Пример задания.** Составить программу движения исполнителя по заданной траектории с двумя датчиками света, используя пропорционально-кубический регулятор.

*Методические рекомендации к заданию:* добавив к пропорциональному регулятору кубический, ученики получают изгиб в функции регулятора

(см. Рисунок 19), что позволит уменьшить раскачивания исполнителя и позволит ему совершать более сильные рывки при удалении от траектории. На тренировочном заезде учащиеся заметят, как увеличилась скорость прохождения исполнителем траектории.

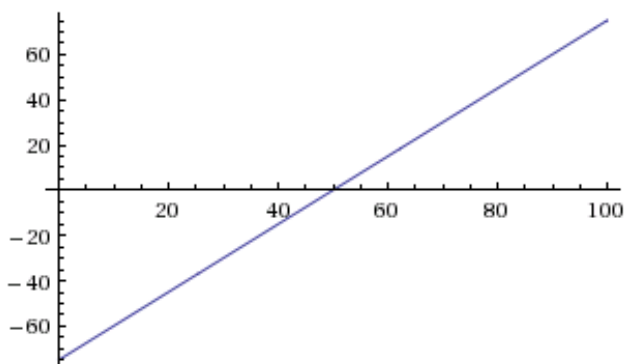


Рисунок 18. График зависимости пропорционально регулятора

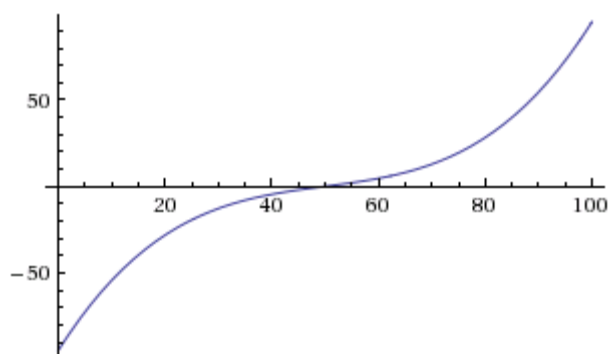


Рисунок 19. График зависимости пропорционально-кубического регулятора

**Пример задания.** Составить программу, заставляющую исполнителя с двумя датчиками света двигаться по заданной траектории.

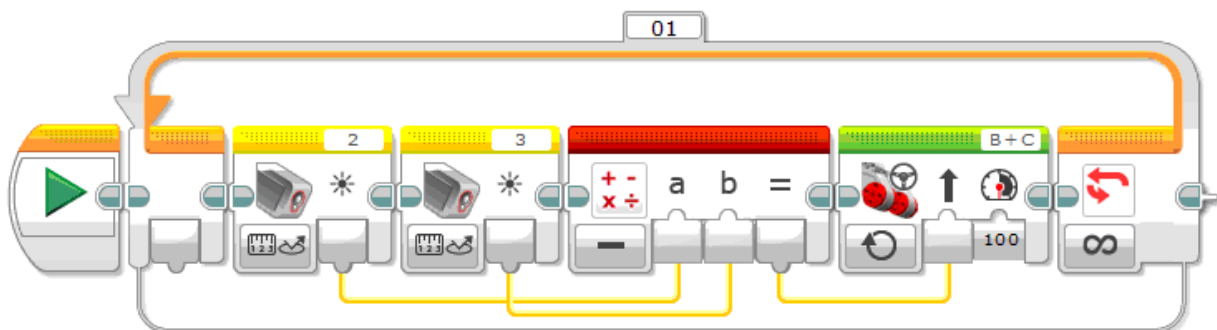


Рисунок 20. Фрагмент программы движения исполнителя с двумя датчиками света

*Методические рекомендации к заданию:* использование двух датчиков света позволяет более чётко разграничить отклонение исполнителя от линии.

**Пример задания.** Составить программу, которая заставит исполнителя двигаться вперед до тех пор, пока инфракрасный датчик не обнаружит, что он находится рядом со стеной или с другим объектом, например, на расстоянии 10см.

**Пример задания.** «В заточении». На белой схеме движения нарисован черный круг. Исполнитель находится в центре круга. Составить программу движения исполнителя внутри черного круга. Исполнитель должен ехать вперед, пока под ним черный цвет и поворачивать вправо, если нет черного цвета. Движение робота должно продолжаться 60 секунд. Программа должна использовать не менее двух циклов.

## Практическое занятие №4-5

## Разработка программы, содержащей оператор ветвления

**Пример задания.** Составить программу движения исполнителя по черной линии (см. Рисунок 21). Исполнитель должен двигаться, отслеживая все ее повороты.

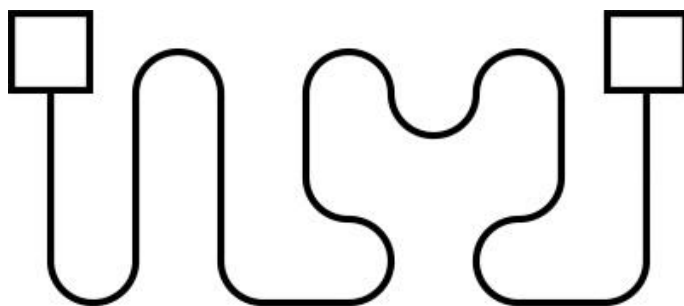


Рисунок 21. Траектория движения для исполнителя

**Пример задания.** Составить программу движения исполнителя по заданной траектории с двумя датчиками света.

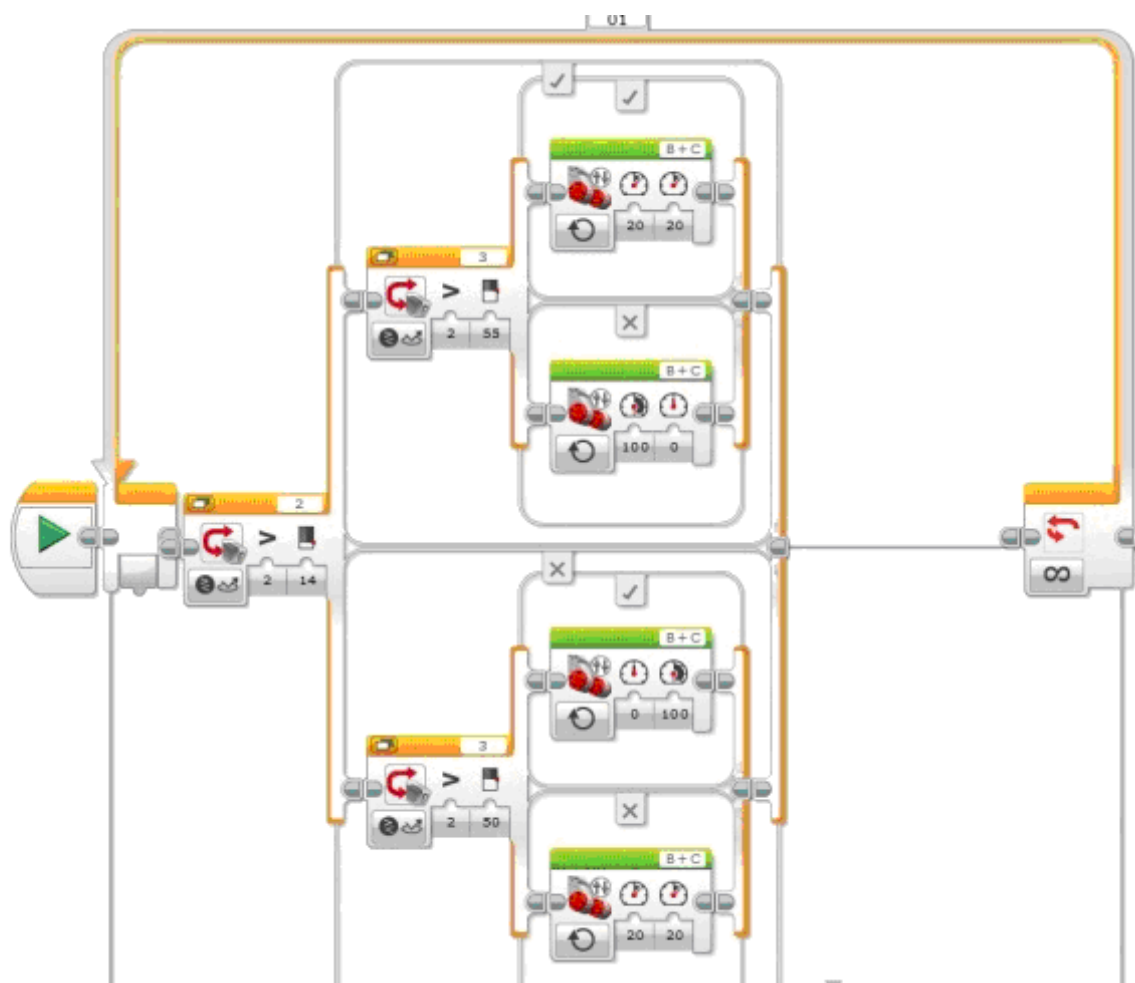
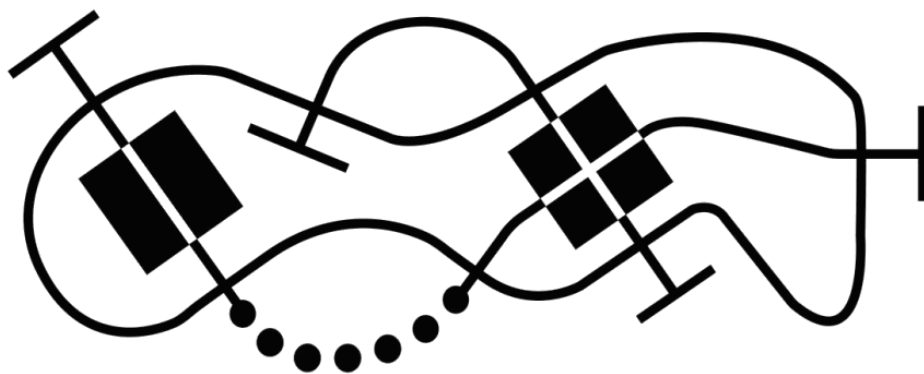


Рисунок 22. Фрагмент программы движения исполнителя по заданной траектории с двумя датчика света

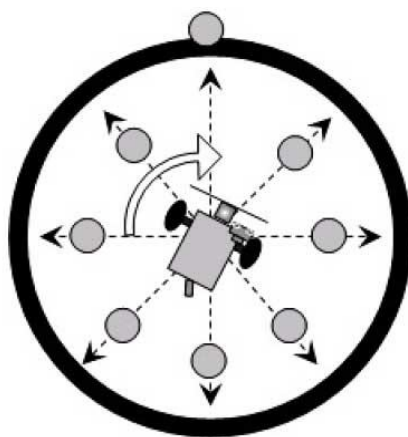
Методические рекомендации к заданию: усложнение решения одной и той же задачи наглядно продемонстрирует различия между существующими алгоритмическими конструкциями, что поспособствует эффективному овладению навыками программирования. Для того чтобы учащиеся увидели эффективность использования при решении задач разных алгоритмических конструкций, необходимо усложнить траекторию движения исполнителя (см. Рисунок 22).

Особенность применения конструкции ветвления для данного задания заключается в следующем: учащимся необходимо провести калибровку датчика одного цвета и определить среднее значение серого. В зависимости от показаний датчиков, то есть при отклонении от линии, исполнитель будет корректировать свой курс.



*Рисунок 23. Пример сложной траектории для движения исполнителя*

**Пример задания.** «Кегельринг». На расстоянии 5см от линии, внутри круга, на равном расстоянии друг от друга стоят пластиковые стаканчики. Исполнитель должен вытолкнуть все стаканчики за пределы круга за наименьшее время.



*Рисунок 24. Траектория движения для исполнителя*

**Пример задания.** «Осторожно, преграда». Составить программу, которая заставит исполнителя постепенно снижать скорость по мере приближения к стене или другому объекту.

**Пример задания.** Составить программу, которая заставляет робота ждать до тех пор, пока не будет нажат ИК-маяк, затем он двигается вперед в течение 20 секунд.

**Пример задания.** «Есть контакт!». Составить программу, которая проверяет, нажат ли датчик касания. Если нажат, то выполняется вариант



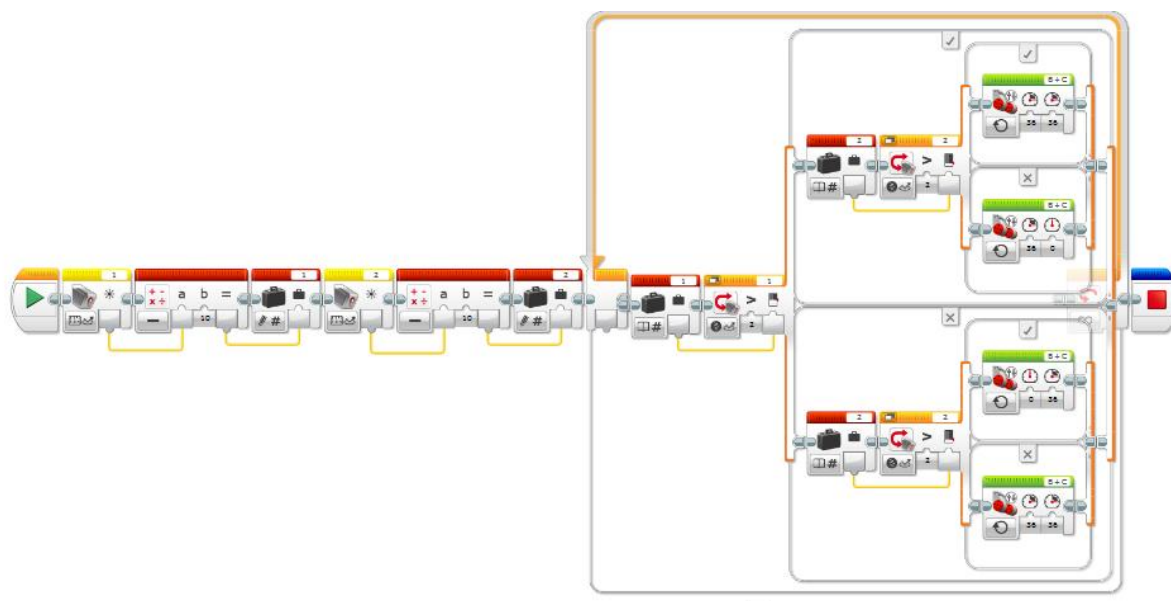
**Практическое занятие №6**

**Алгоритмические конструкции: следование, ветвление, повторение**

**Пример задания.** «Правила дорожного движения». Составить программу, позволяющую исполнителю отслеживать перекрестки и суммировать их количество.

## Алгоритмические конструкции: следование, ветвление, повторение

**Пример задания.** «Правила дорожного движения». Составить программу, позволяющую исполнителю отслеживать перекрестки и суммировать их количество.



*Методические рекомендации к заданию:* данная задача может являться сюжетной, например, при формулировке задания, можно провести для учащихся аналогию с правилами дорожного движения. Для усложнения задания, с целью повышения мотивации учащихся к работе, рекомендуется предложить изменить задачу так, чтобы исполнитель останавливался при обнаружении перекрестка на 2 секунды, как бы проверяя приближение транспортных средств.



**Пример задания.** «Лабиринт». Используя подпрограмму универсального движения по траектории, составить программу эффективного прохождения исполнителем лабиринта.

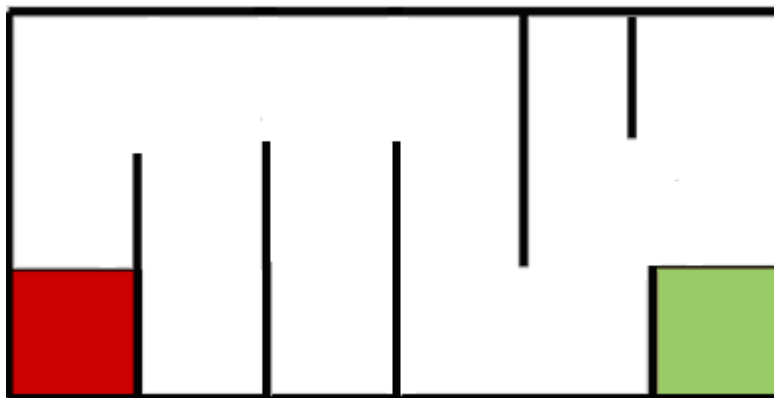


Рисунок 27. Лабиринт для исполнителя

### Практическое занятие №8-9

#### Разработка программы по обработке одномерного массива

**Пример задания.** «Лабиринт». Составить программу, заставляющую исполнителя найти выход из лабиринта за указанное количество шагов. Конструкция устройства предусматривает один гироскопический датчик для коррекции поворота.

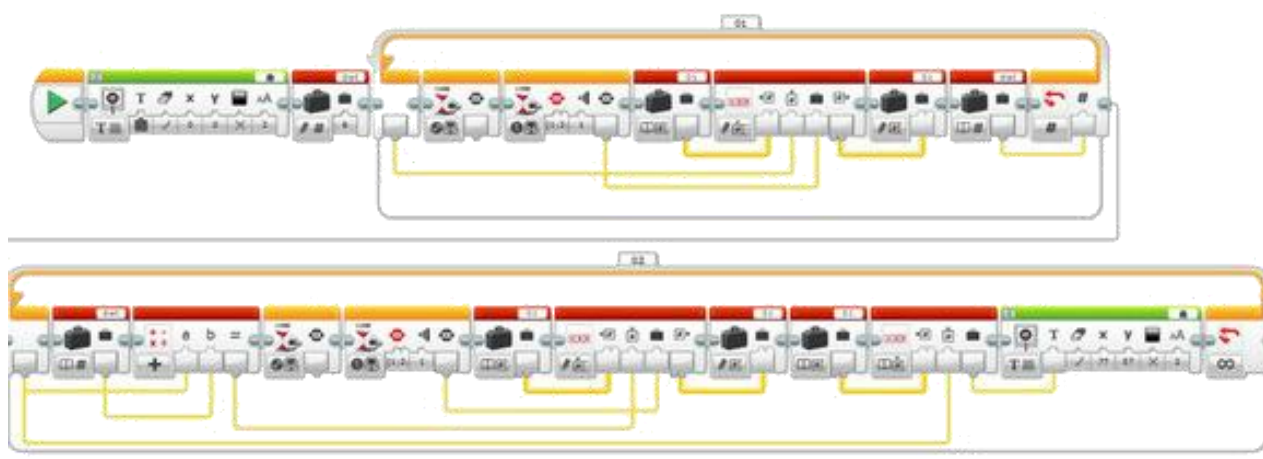


Рисунок 28. Фрагмент программы прохождения исполнителем лабиринта

**Пример задания.** «Лабиринт». Составить программу, заставляющую исполнителя найти выход из лабиринта за указанное количество шагов, а затем вернуться обратно в стартовую точку.

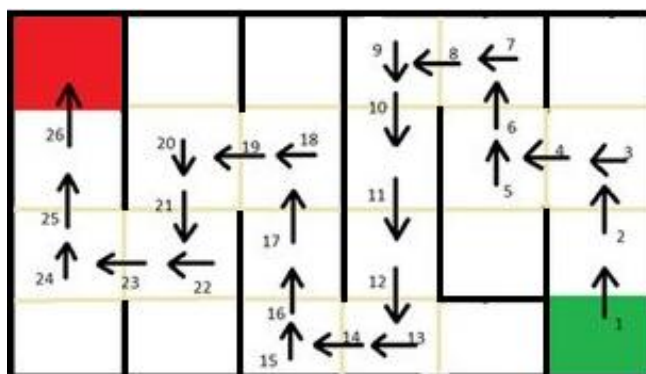


Рисунок 29. Лабиринт для исполнителя

**Пример задания.** «Лабиринт». Составить программу, заставляющую исполнителя найти выход из лабиринта за наименьшее количество шагов, а затем этим же путем вернуться обратно в стартовую точку.

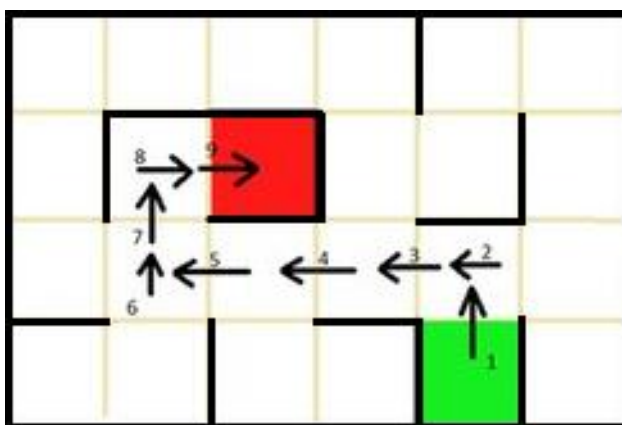


Рисунок 30. Лабиринт для исполнителя

## Практическое занятие №10

### Создание алгоритма, решающего поставленную задачу

На последнем занятии целесообразно предложить учащимся сконструировать и запрограммировать устройство, решающее свою, неповторимую задачу.

Примеры проектов:

- «Инстинкт самосохранения» – реакция исполнителя на громкость звука: увеличение скорости, проявление эмоций;
- «Автоответчик» – воспроизведение исполнителем звукового файла при голосовом обращении к нему;

- «Домашний шумометр» – отображение на экране графика изменения звуковой обстановки вокруг исполнителя;
- «Дневной автомобиль» – изменение скорости движения робота в зависимости от освещенности;
- «Безопасный автомобиль» – движение робота с фиксированной скоростью или в зависимости от освещенности помещения по условию;
- «Измеритель освещенности» – отображение на экране графика освещенности окружающей обстановки с поворотом координатной плоскости.

### ***2.3. Апробация разработанных материалов***

Апробация разработанных материалов проводилась методом экспертных оценок. Экспертами выступали студенты Уральского государственного педагогического университета Института математики, информатики и информационных технологий, в количестве 10 человек.

Экспертам были представлены методические рекомендации и содержание уроков-практик. Оценить разработанные материалы эксперты могли, отвечая на заранее подготовленные вопросы. Целью анкет было оценить качество и возможность практического применения результатов исследования.

Анкета 1.

Экспертная оценка разработанных методических материалов

1. Соответствуют ли разработанные материалы выбранному разделу «Алгоритмизация и программирование»?
  - а) полностью соответствуют;
  - б) частично соответствуют;
  - с) не соответствуют.
2. Насколько интересны разработанные материалы?
  - а) очень интересны;

- b) интересны;
  - c) не интересны.
3. Насколько эффективны разработанные материалы для внедрения в образовательный процесс?
- a) эффективны;
  - b) частично эффективны;
  - c) не эффективны.
4. Соответствуют ли разработанные материалы требованиям ФГОС к формированию УУД?
- a) полностью соответствуют;
  - b) частично соответствуют;
  - c) не соответствуют.
5. Стали бы вы использовать разработанные задания в своей педагогической деятельности?
- a) использовал(а) бы полностью;
  - b) использовал(а) бы частично;
  - c) не использовал(а) бы.

### Результаты апробации

Результаты апробации представлены в виде диаграмм, сформированным по результатам обработки оценок экспертов.

Диаграмма 1.

Соответствуют ли разработанные материалы выбранному разделу «Алгоритмизация и программирование»?

(10 ответов)

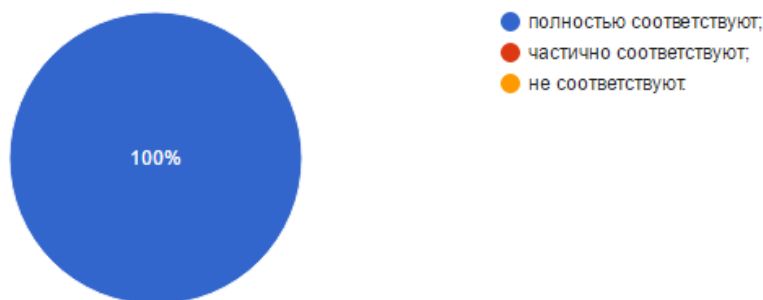


Диаграмма 2.

Насколько интересны разработанные материалы? (10 ответов)

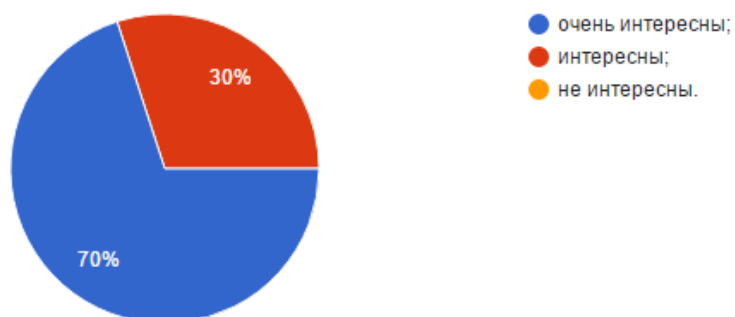


Диаграмма 3.

Насколько эффективны разработанные материалы для внедрения в образовательный процесс? (10 ответов)

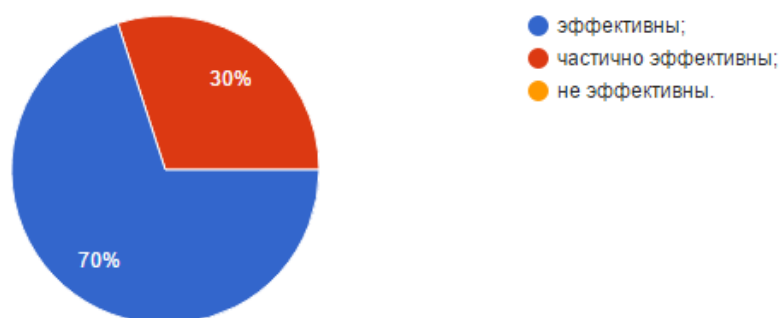


Диаграмма 4.

Соответствуют ли разработанные материалы требованиям ФГОС к формированию УУД? (10 ответов)

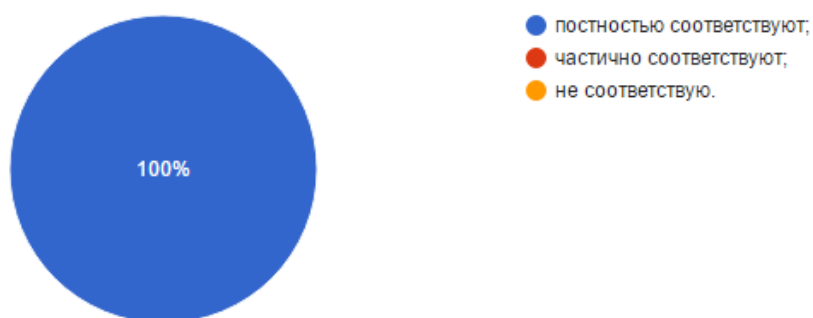
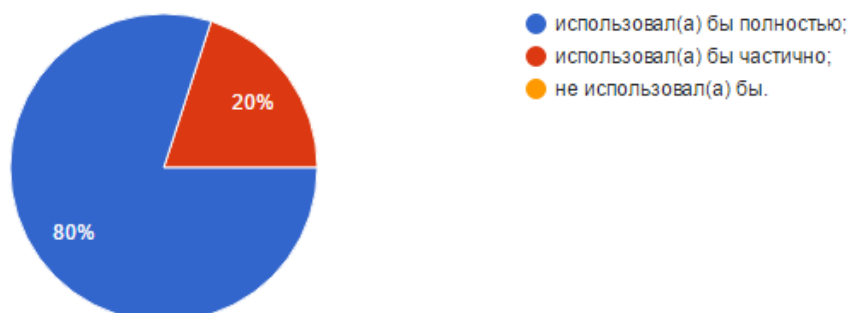


Диаграмма 5.

Стали бы вы использовать разработанные задания в своей педагогической деятельности?

(10 ответов)





## **Заключение**

Для реализации требований ФГОС перед учителями встает задача поиска новых методических средств и приемов, главной особенностью которых является направленность на овладение универсальными учебными действиями. Обучающиеся должны самостоятельно научиться ставить цели и определять пути их достижения, использовать приобретенный в школе опыт за рамками учебного процесса – в реальной жизни.

В рамках дисциплины «Информатика и ИКТ» реализовать поставленную задачу позволяет использование в учебном процессе реальных исполнителей с обратной связью. Такие исполнители поощряют детей мыслить творчески, анализировать ситуацию и применять критическое мышление для решения реальных проблем. Работа в команде и сотрудничество укрепляет коллектив, а соперничество на соревнованиях дает стимул к учебе. Использование исполнителей с обратной связью приучит учащихся смотреть на проблемы шире и решать их в комплексе. Сконструированная модель при этом всегда находит аналог в реальном мире. Задачи, которые ученики ставят исполнителю, предельно конкретны, а в процессе создания управляющих алгоритмов обнаруживаются ранее непредсказуемые свойства устройства или открываются новые возможности его использования.

В ходе работы были изучены требования ФГОС к объему и содержанию раздела «Алгоритмизация и программирование» и рассмотрены существующие учебно-методические комплексы, отвечающие данным требованиям. Был проведен анализ используемых при обучении виртуальных и реальных исполнителей. На основе анализа сделан вывод о том, что наборы LEGO MINDSTORMS Education EV3 наиболее адаптированы для внедрения в учебный процесс.

В практической части исследования разработаны методические рекомендации по использованию реальных исполнителей в рамках содержательной линии «Алгоритмизация и программирование» дисциплины

«Информатика и ИКТ». В соответствии с разработанными рекомендациями проработано примерное содержание уроков-практик. Проведена апробация разработанных материалов.

По результатам апробации, разработанные методические материалы соответствуют выбранному разделу дисциплины, а их практическое использование в учебном процессе будет интересным и эффективным. Исходя из этого, можно сделать вывод о том, что посредством успешного решения задач в ходе исследовательской работы, поставленная цель была достигнута.

## Библиографический список

1. Гейн А.Г., Юнерман Н.А., Гейн А.А. Информатика, Учебник для общеобразовательных учреждений, 9 класс. – М.: Просвещение, 2012. – 191 с.
2. Доронина К.Е., Шимов И.В. Особенности использования реальных исполнителей при обучении программированию // Актуальные вопросы преподавания математики, информатики и информационных технологий: межвузовский сборник научных работ. – Екатеринбург: Урал. гос. пед. ун-т, 2016. – С. 136-144.
3. Доронина К.Е., Шимов И.В. Использование исполнителей с обратной связью при изучении алгоритмизации на уроках информатики в школе // Актуальные вопросы преподавания математики, информатики и информационных технологий: межвузовский сборник научных работ. – Екатеринбург: Урал. гос. пед. ун-т, 2015. – С. 159-163.
4. Единая коллекция ЦОР URL: <http://school-collection.edu.ru/> (дата обращения: 17.12.15).
5. Информатика. Комплект Макаровой Н.В. // Учебно-методический комплект по информатике и ИКТ Макаровой Н.В. URL: <http://makarova.piter.com/> (дата обращения: 17.12.15).
6. Информатика и ИКТ: Практикум. 8-9 класс / Под ред. проф. Макаровой Н.В. – СПб.: Питер, 2010. – 384 с.
7. Крицын А.А. Основные тенденции развития образовательной робототехники // Научный сборник «Современная школа России. Вопросы модернизации». Материалы международной научно-практической конференции. – М.: 2012. – С. 6-65.
8. Кушниренко А.Г., Лебедев Г.В., Зайдельман Я.Н. Информатика. 7-9 класс: Учеб. для общеобразоват. учеб. заведений. – 3-е изд., стереотип. – М.: Дрофа, 2002. – 464 с.

9. Макарова Н.В. Методическое пособие для учителя. Программа по информатике и ИКТ 5-11 класс. – СПб.: ПИТЕР, 2012. – 86 с.
10. Пейперт С. Переворот в сознании: дети, компьютеры и плодотворные идеи. – М.: Педагогика, 1989. – 220 с.
11. Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В. Учебник «Информатика» для 9 класса. – М.: БИНОМ. Лаборатория знаний, 2011. – 334 с.
12. Семакин И.Г. Информатика. УМК для основной школы 7-9 классы. Методическое пособие для учителя. – М.: БИНОМ. Лаборатория знаний, 2013. – 184 с.
13. Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В. Программа основного общего образования по информатике, 7-9 класс / – М.: «Издательство БИНОМ. Лаборатория знаний», 2013. – 576 с.
14. Стандарт основного общего образования по информатике и ИКТ (из приложения к приказу Минобробразования России от 05.03.04 №1089) / Программы для общеобразовательных учреждений. Информатика. 2-11 классы: методическое пособие – М.: БИНОМ. Лаборатория знаний, 2012.
15. Угринович Н.Д., Информатика и ИКТ: учебник для 9 класса – М.: БИНОМ. Лаборатория знаний, 2012. – 295 с.
16. Угринович Н.Д., Самылкина Н.Н. Информатика. Программа для основной школы 7-9 класс. – М.: БИНОМ. Лаборатория знаний, 2012. – 53 с.
17. Угринович Н.Д. Информатика. УМК для основной школы 7-9 классы. Методическое пособие для учителя. – М.: БИНОМ. Лаборатория знаний, 2013. – 91 с.
18. Учебник «Информатика» для 8-9 классов / Макарова Н.В., Волкова И.В., Николайчук Г.С. и др., Под ред. Макаровой Н.В. – СПб.: Питер-Пресс, 2012. – 418 с.

19. Федеральный государственный образовательный стандарт основного общего образования // Министерство образования и науки Российской Федерации URL: <http://минобрнауки.рф/документы/543> (дата обращения: 23.11.15).
20. Цветкова М.С. ФГОС. Информатика. Основная школа // Интернет-газета «Лаборатория знаний» издательства БИНОМ. – 2012. – Сентябрь.
21. Шимов И.В. Применение робототехнических устройств в обучении программированию школьников // Педагогическое образование в России. – 2013. – С. 185-188.
22. LEGO Education. Продукция. URL: <http://education.lego.com/ru-ru/products>. (дата обращения: 17.12.15).
23. LEGO: больше чем конструктор // Занимательная робототехника URL: <http://edurobots.ru/2014/11/lego-bolshe-chem-konstruktor/> (дата обращения: 23.03.16).
24. Robotics. URL: <https://ru.wikipedia.org/wiki/Robotics> (дата обращения: 13.01.16).
25. Fischertechnik. URL: <https://ru.wikipedia.org/wiki/Fischertechnik> (дата обращения: 13.01.16).

## Приложения

### Приложение 1



*Рисунок 31. Модель исполнителя с двумя датчиками света*

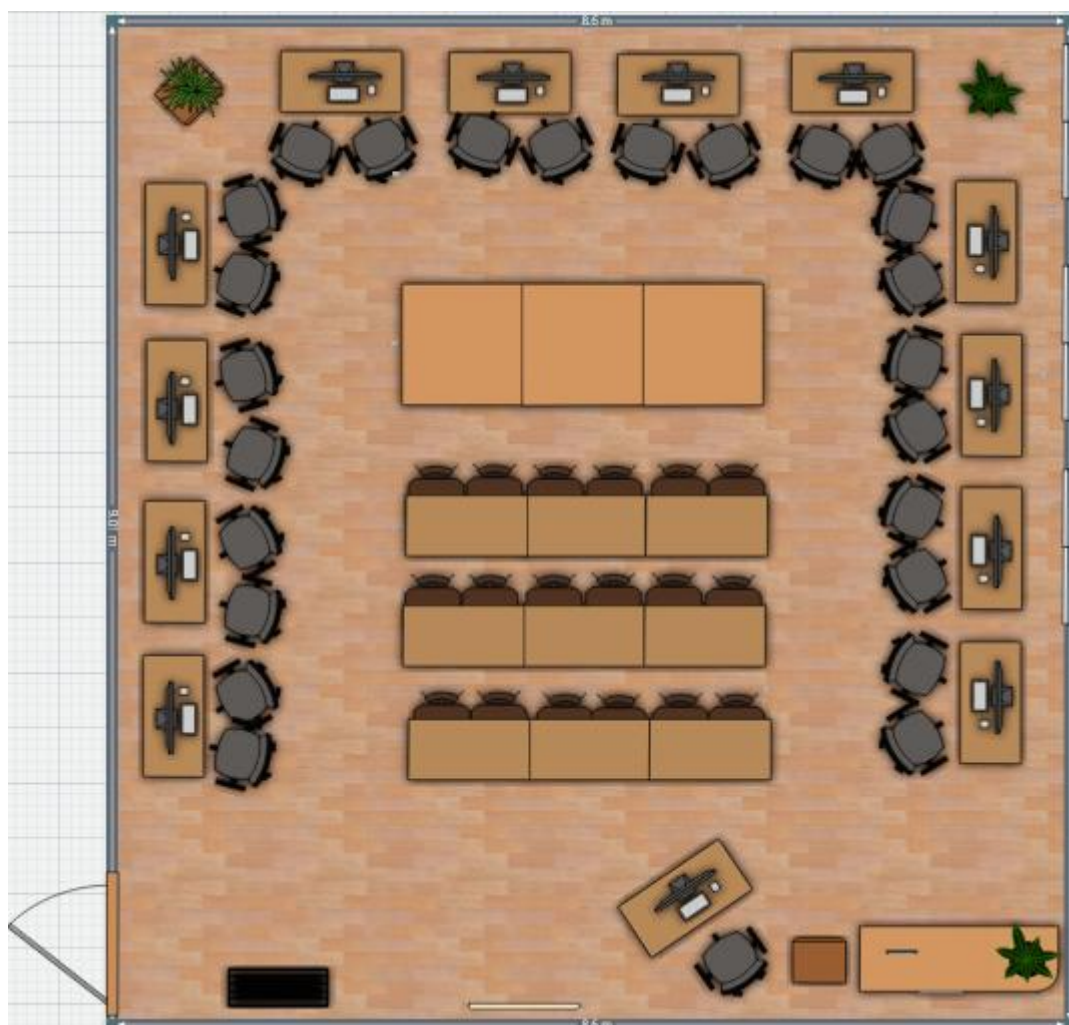
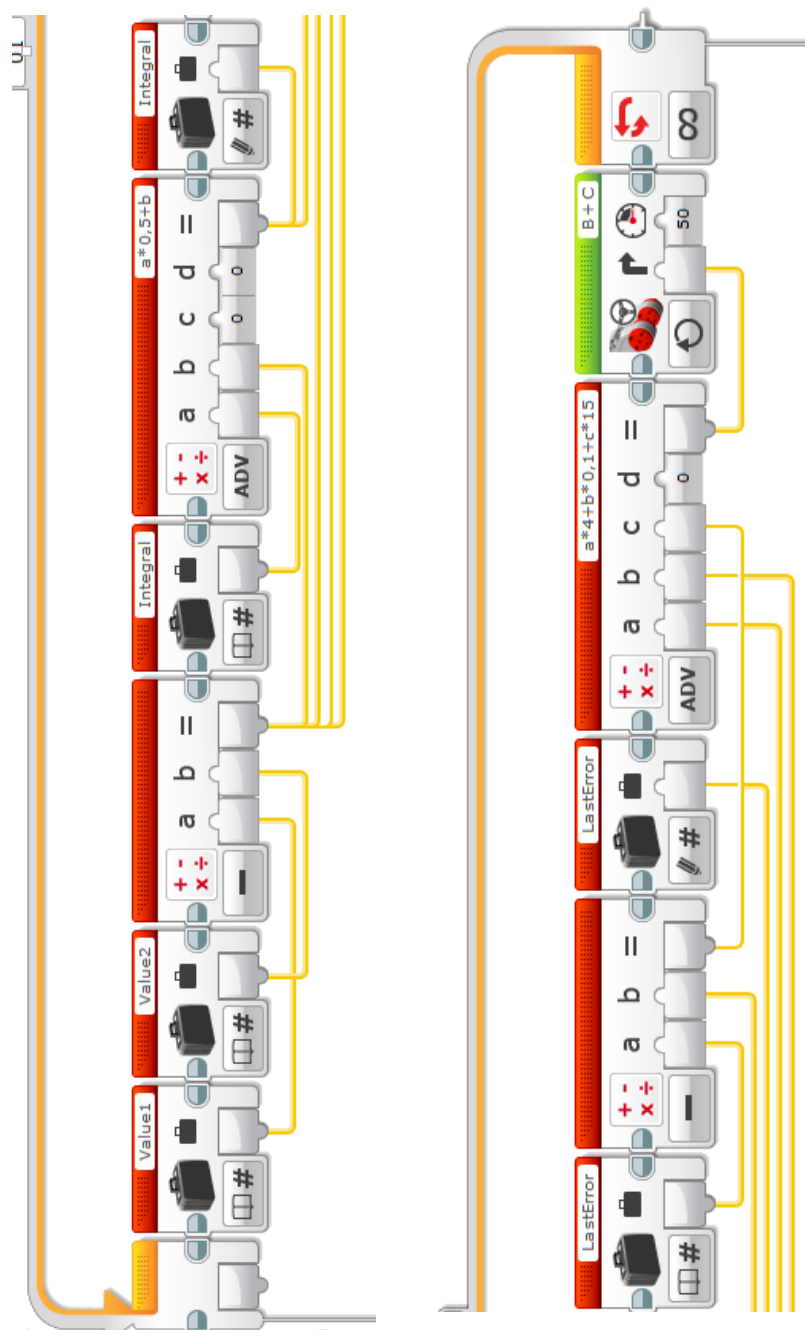


Рисунок 32. Схема кабинета информатики и ИКТ



Рисунок 33. Реализация универсального алгоритма пропорционального дифференциального интегрального регулятора (начало)



*Рисунок 34. Реализация универсального алгоритма пропорционального дифференциального интегрального регулятора*  
(продолжение)